

FAU-TP3-97/1  
NIKHEF-97/013  
Feb. 27, 1997

# Stochastic Methods for Zero Energy Quantum Scattering

Justus H. Koch<sup>a,b</sup>, Hubertus R. Mall<sup>a</sup>, Stefan Lenz<sup>a</sup>

<sup>a</sup>Institut für Theoretische Physik III, Universität Erlangen - Nürnberg  
Staudtstraße 7, D-91058 Erlangen, Germany

<sup>b</sup>National Institute for Nuclear Physics and High Energy Physics (NIKHEF)  
Postbus 41882, 1009 DB Amsterdam, The Netherlands  
and Institute for Theoretical Physics, University of Amsterdam  
email: justus@nikhef.nl

## Abstract

We investigate the use of stochastic methods for zero energy quantum scattering based on a path integral approach. With the application to the scattering of a projectile from a nuclear many body target in mind, we use the potential scattering of a particle as a test for the accuracy and efficiency of several methods. To be able to deal with complex potentials, we introduce a path sampling action and a modified scattering observable. The approaches considered are the random walk, where the points of a path are sequentially generated, and the Langevin algorithm, which updates an entire path. Several improvements are investigated. A cluster algorithm for dealing with scattering problems is finally proposed, which shows the best accuracy and stability.

# 1 Introduction

The most frequently used method to study small scale composite systems is quantum scattering. While the interaction of the projectile with an isolated constituent of the target may be well known and can be described accurately, the scattering from the bound system can in general only be solved with several simplifying assumptions. These approximations concern both the interaction of the projectile with the target as well as the internal target dynamics during the scattering process.

While a better understanding of the scattering from composite systems is of interest for several fields of physics, we focus here specifically on the scattering of a strongly interacting particle from nuclei at zero energy. There exist several often used recipes for the description of low energy nuclear scattering, mainly in the form of effective single particle potentials for the projectile - nucleus interaction. However, the inherent approximations have not been tested in a conclusive way. That there are surprises if one goes beyond the standard approximations has been shown recently in the framework of the frozen nucleus picture, where all nucleons are held fixed during the scattering process. By evaluating the scattering amplitude with Monte Carlo methods for a target with a large number of scattering centers, it was found that the projectile gets 'trapped' by clusters of target constituents [1], [2]. This is an effect which is absent in any of the approximate descriptions. Clearly it is important to investigate methods how to solve in practice the scattering from a many-body target exactly, such that also the target dynamics can be taken into account.

A formalism that has hardly ever been applied to scattering is the path integral description. In a recent study [3] this approach was applied to the potential scattering of a particle and stochastic methods were used to evaluate the multi-dimensional integrals one encounters. It was then shown that these methods can in principle be extended in a straightforward fashion to calculate the exact scattering amplitude for scattering from nuclei with the full inclusion of the target dynamics. Before this method can be applied to a detailed study of nuclear scattering, which requires a considerable computational effort, it is necessary to find more efficient and numerically stable stochastic methods to obtain the scattering amplitude. This is the purpose of this paper where we examine different stochastic algorithms that can be applied to the scattering problem. To be able to assess the accuracy of these methods we apply them to the potential scattering of a low energy particle, where the exact solution can be obtained by standard numerical methods from the single

particle Schrödinger equation. Keeping in mind the application to problems such as the interactions of low energy anti-protons with nuclei, we include complex potentials in our investigation of stochastic methods for scattering, which has not been considered before.

In several fields of physics path integrals are a well established approach to obtain the ground state properties of a system and there exist stochastic methods to evaluate the resulting high-dimensional integrals. Examples are spin systems [4], [5] and quantum fields on the lattice [6]. However, the subject of these studies are localized, bound systems. The application of these methods to scattering, where the states are not normalizable and the spectrum is continuous, is therefore not straightforward. Another complication is due to the use of complex potentials, which have been already used for bound states *e.g.* in quantum chemistry [7, 8] or in the calculation of correlation functions of nuclear ground states [9]. A complex action makes it impossible to interpret the distribution of paths as a probability distribution and we examine how to deal with this problem in scattering. For completeness we mention here also the possibility of obtaining the scattering length of a system from the volume dependence of its energy spectrum [10]. The feasibility of this approach has been shown for some special cases of quantum field theory on the lattice: the non-linear  $\sigma$  model in two dimensions [11] and the Ising model in four dimensions [12]. The generalization of this method to nuclear scattering is certainly not straightforward and we therefore focus below on more direct path integral methods.

In Section 2, we briefly outline the path integral approach to non-relativistic quantum scattering and define a 'modified observable' appropriate for scattering calculations with a complex action. Different stochastic methods to evaluate the resulting multi-dimensional integrals are discussed in the subsequent chapters. We examine their efficiency and accuracy in order to identify the method that is most suitable for the application to many-body targets. The random walk algorithm, the most straightforward implementation of the path integral concept, is discussed in Section 3. Langevin and hybrid methods, which update entire paths, are studied in Section 4. An efficient new approach, a cluster algorithm for quantum mechanical scattering, is developed in Section 5. A summary and our conclusions are presented in Section 6.

## 2 Potential Scattering

### 2.1 Scattering formalism

In this chapter we briefly describe the theoretical framework for solving a scattering problem by using the path integral approach. We discuss only the potential scattering for a single particle. It is used in this paper as a test case for finding a numerical method that is suitable for applications to scattering from a composite many-particle target. Some modifications of standard methods in the literature are necessary since we allow for complex potentials.

The Hamiltonian, which will be used throughout this paper, describes the non-relativistic scattering of a particle from an complex, local potential

$$\hat{H} = \frac{\hat{p}^2}{2\mu} + \hat{V} = \hat{K} + \hat{V}, \quad (1)$$

$$\hat{V} = \hat{U} + i \hat{W}, \quad (2)$$

where  $\hat{p}$  is the momentum operator and  $\mu$  the reduced mass of projectile - target system. We are interested in stationary scattering states, corresponding to a real energy  $E(k) = \frac{\vec{k}^2}{2\mu}$ . Due to the complex potential the Hamiltonian  $\hat{H}$  is not hermitian. Therefore one has two solutions to the Lippmann-Schwinger equations, which belong to  $\hat{H}$  and  $\hat{H}^\dagger$ , respectively [13]:

$$\begin{aligned} |\psi_k^+\rangle &= |\chi_k\rangle + \frac{1}{E^+ - \hat{H}} \hat{V} |\chi_k\rangle, & |\psi_k^-\rangle &= |\chi_k\rangle + \frac{1}{E^- - \hat{H}^\dagger} \hat{V}^\dagger |\chi_k\rangle, \\ |\tilde{\psi}_k^-\rangle &= |\chi_k\rangle + \frac{1}{E^- - \hat{H}} \hat{V} |\chi_k\rangle, & |\tilde{\psi}_k^+\rangle &= |\chi_k\rangle + \frac{1}{E^+ - \hat{H}^\dagger} \hat{V}^\dagger |\chi_k\rangle. \end{aligned} \quad (3)$$

We assume the existence of a complete biorthogonal basis for the Hamiltonian,  $\hat{H}$  [14], [15].

The scattering wave functions  $|\hat{\psi}_0^\pm\rangle$  for zero projectile energy can be obtained from the propagator  $\hat{\rho}(\beta)$ ,

$$\hat{\rho}(\beta) = \frac{1}{\mathcal{N}_\beta} \exp[-\beta \hat{H}], \quad (4)$$

$$\mathcal{N}_\beta = \left( \frac{2\pi\mu}{\beta} \right)^{\frac{3}{2}}, \quad (5)$$

in the limit of large Euclidean times  $\beta$ . This operator contains the complete physical information about the evolution of the system. Its spectral representation in the biorthogonal basis given by Eq. (3) is

$$\hat{\rho}(\beta) = \frac{1}{\mathcal{N}_\beta} \int d^3k \exp[-\beta E(k)] |\psi_k^\pm\rangle \langle \tilde{\psi}_k^\pm|. \quad (6)$$

In the limit  $\beta \rightarrow \infty$  one is left with

$$\lim_{\beta \rightarrow \infty} \hat{\rho}(\beta) = |\psi_0^\pm\rangle \langle \tilde{\psi}_0^\pm|, \quad (7)$$

which projects onto the ground state of the system. This projector can be used to calculate the *square* of the scattering amplitude,

$$f_0^2 = \left[ (2\pi)^2 \mu \right]^2 \langle \chi_0 | \hat{V} | \psi_0^+ \rangle \langle \tilde{\psi}_0^- | \hat{V} | \chi_0 \rangle. \quad (8)$$

The wavefunctions  $|\chi_0\rangle$  are plane wave solutions, which in the zero energy limit are constant. The wave functions  $|\tilde{\psi}_k^+\rangle$  and  $|\psi_k^-\rangle$  differ with respect to their boundary conditions. However, since no spatial direction is distinguished in the limit  $E(k) \rightarrow 0$ , Eq. (8) therefore can be written as

$$\begin{aligned} f_0^2 &= \left[ (2\pi)^2 \mu \right]^2 \langle \chi_0 | \hat{V} | \psi_0^+ \rangle \langle \tilde{\psi}_0^+ | \hat{V} | \chi_0 \rangle \\ &= \left[ (2\pi)^2 \mu \right]^2 \lim_{\beta \rightarrow \infty} \langle \chi_0 | \hat{V} \hat{\rho}(\beta) \hat{V} | \chi_0 \rangle, \end{aligned} \quad (9)$$

or in  $r$ -space representation

$$f_0^2 = \left[ (2\pi)^2 \mu \right]^2 \lim_{\beta \rightarrow \infty} \int d\vec{r}_1 d\vec{r}_{N+1} \chi_0(\vec{r}_{N+1}) V(\vec{r}_{N+1}) \rho(\vec{r}_{N+1}, \vec{r}_1; \beta) V(\vec{r}_1) \chi_0(\vec{r}_1). \quad (10)$$

Note that, in contrast to other discussions of scattering in the literature, the quantity  $f_0^2$  is calculated instead of  $|f_0|^2$ . It contains information not only about the absolute value, but also the phase of the zero energy scattering amplitude.

In the following chapters, we will investigate different stochastic methods to solve Eq. (10). In all subsequent applications, the shape of  $\hat{V}$  will be assumed to be Gaussian,

$$V(\vec{r}) = (U_0 + iW_0) v(\vec{r}) \equiv V_c v(\vec{r}), \quad (11)$$

$$v(\vec{r}) = \exp\left(-\frac{1}{2b^2} \vec{r}^2\right), \quad (12)$$

but this assumption is not crucial for the methods discussed here. Keeping in mind applications to the scattering length of antiprotons [16], [17], we take the real part of

the potential  $U_0$  in the range of  $-0.5 \text{ fm}^{-1}$  to  $2.5 \text{ fm}^{-1}$  and  $W_0$  from 0 to  $-7.0 \text{ fm}^{-1}$ . The width parameter will always be taken to be  $b = 0.5 \text{ fm}$  and the reduced mass  $\mu = 2.5 \text{ fm}^{-1}$ .

## 2.2 Path integral approach and modified observable

For the actual evaluation of Eq. (9) we use the Feynman path integral approach. For this purpose the projector, Eq. (4), is rewritten as [18]

$$\rho_{\mathcal{A}}(\vec{r}_{N+1}, \vec{r}_1; \beta) \approx \frac{1}{\mathcal{N}_{\beta} \mathcal{N}_{\varepsilon}^N} \prod_{n=2}^N \left[ \int_{-\infty}^{\infty} d\vec{r}_n \right] \exp \left[ -\mathcal{A}[\vec{R}] \right], \quad (13)$$

$$\mathcal{A}[\vec{R}] = \sum_{n=2}^{N+1} \left[ \frac{\mu}{2\varepsilon} (\vec{r}_n - \vec{r}_{n-1})^2 + \frac{\varepsilon}{2} (V_n + V_{n-1}) \right], \quad (14)$$

$$\mathcal{N}_{\varepsilon} = \left[ 2\pi \frac{\varepsilon}{\mu} \right]^{\frac{3}{2}}, \quad \beta = \varepsilon N. \quad (15)$$

In the integrand of the path integral  $\mathcal{A}[\vec{R}]$  denotes the Euclidean action and  $\vec{R} = [\vec{r}_1, \vec{r}_2, \dots, \vec{r}_N]$  a path consisting of  $N$  steps. Each step corresponds to an increment  $\varepsilon$  of the imaginary time,  $\beta$ . Parts of the potentials appearing in Eq. (9) will be included into the projector by extending the action to

$$\tilde{\mathcal{A}}[\vec{R}] \equiv \mathcal{A}[\vec{R}] - \ln v_1 - \ln v_{N+1}, \quad v_n \equiv v(\vec{r}_n), \quad (16)$$

or in our case to

$$\tilde{\mathcal{A}}[\vec{R}] = \mathcal{A}[\vec{R}] + \frac{1}{2b^2} [\vec{r}_1^2 + \vec{r}_{N+1}^2]. \quad (17)$$

Combining different factors into

$$\mathcal{N} \equiv \frac{\mathcal{N}_{\beta} \mathcal{N}_{\varepsilon}^N}{2\pi\mu^2}, \quad (18)$$

and with the definitions

$$P_{\mathcal{A}}[\vec{R}] = \exp \left[ -\tilde{\mathcal{A}}[\vec{R}] \right] \quad (19)$$

and

$$\mathcal{D}[\vec{R}] = d\vec{r}_1 d\vec{r}_2 \dots d\vec{r}_{N+1}, \quad (20)$$

the expression for the scattering amplitude, Eq. (9), finally becomes a multi-dimensional integral

$$f_0^2 = \frac{V_c^2}{\mathcal{N}} \int \mathcal{D}[\vec{R}] P_{\mathcal{A}}[\vec{R}]. \quad (21)$$

The integral now also includes the endpoints  $\vec{r}_1$  and  $\vec{r}_{N+1}$  of the path.

The expression for the scattering amplitude, Eq. (21), contains only known quantities, *e.g.* it does not involve the unknown scattering wave function. Instead one has to deal with multi-dimensional path integrals. In the next chapters we will examine different stochastic methods to evaluate these integrals with sufficient accuracy and speed, such that the path integral approach can be also be applied to many-body targets. A problem is that the potential appearing in the action  $\mathcal{A}$  is complex. The expression in Eq. (21) therefore cannot be interpreted as an integral over paths sampled according to a real and positive distribution, the starting point for many stochastic methods. Even for real potentials, which have both attractive and repulsive regions, this interpretation requires special care [3].

To remedy this difficulty associated with complex potentials, we introduce a suitable real action,  $\tilde{\mathcal{S}}$ , that yields a real positive distribution,

$$P_S[\vec{R}] \equiv \exp[-\tilde{\mathcal{S}}[\vec{R}]], \quad P_S[\vec{R}] \geq 0. \quad (22)$$

The action  $\tilde{\mathcal{S}}$  should simulate the main features of the actual problem as closely as possible or, equivalently,  $P_S[\vec{R}]$  provide a good approximation to the kernel  $P_A[\vec{R}]$  in Eq. (21). For a many-body target, for example, such a reference problem could be described in terms of a simple effective one body potential [3]: in regions where the actual interaction is strongly absorptive and the wavefunction is small, a real reference potential  $U_a(\vec{r})$  may be chosen that is strongly repulsive. Using this approximate reference action, Eq. (21) can be rewritten as

$$f_0^2 = \frac{V_c^2}{\mathcal{N}} \int \mathcal{D}[\vec{R}] P_A[\vec{R}] = \frac{V_c^2}{\mathcal{N}} \int \mathcal{D}[\vec{R}] \left[ \frac{P_A[\vec{R}]}{P_S[\vec{R}]} \right] P_S[\vec{R}]. \quad (23)$$

The expression has the form of a multi-dimensional integral over a generalized complex observable,  $\mathcal{O}[\vec{R}]$ , where the distribution of paths is given by  $P_S[\vec{R}]$ :

$$f_0^2 = \frac{V_c^2}{\mathcal{N}} \int \mathcal{D}[\vec{R}] \mathcal{O}[\vec{R}] P_S[\vec{R}] \equiv \frac{V_c^2}{\mathcal{N}} \langle \mathcal{O} \rangle_{P_S}, \quad (24)$$

$$\mathcal{O}[\vec{R}] = \left[ \frac{P_A[\vec{R}]}{P_S[\vec{R}]} \right], \quad (25)$$

or more explicitly

$$f_0^2 = \frac{V_c^2}{\mathcal{N}} \int \mathcal{D}[\vec{R}] e^{-(\tilde{\mathcal{A}}[\vec{R}] - \tilde{\mathcal{S}}[\vec{R}])} e^{-\tilde{\mathcal{S}}[\vec{R}]} = \frac{V_c^2}{\mathcal{N}} \langle e^{-\Delta\mathcal{S}} \rangle_{P_S}, \quad (26)$$

$$(27)$$

where

$$\Delta\mathcal{S}[\vec{R}] = \tilde{\mathcal{A}}[\vec{R}] - \tilde{\mathcal{S}}[\vec{R}] . \quad (28)$$

As we will see, in working with complex modified observables one encounters strong numerical fluctuations which are caused by the presence of a phase that varies along the entire path.

Monte Carlo methods used to generate paths distributed according to  $P_S$  always yield a normalized distribution. Except for few cases, this normalization is not known. Therefore, instead of Eq. (26) one calculates the quotient

$$\frac{f_0^2}{f_{0a}^2} = \frac{V_c^2}{U_a^2} \frac{\int \mathcal{D}[R] e^{-[\tilde{\mathcal{A}}[\vec{R}] - \tilde{\mathcal{S}}[\vec{R}]]} e^{-\tilde{\mathcal{S}}[\vec{R}]}}{\int \mathcal{D}[R] e^{-\tilde{\mathcal{S}}[\vec{R}]}} \quad (29)$$

$$= \frac{V_c^2}{U_a^2} \frac{\int \mathcal{D}[R] e^{-\Delta\mathcal{S}[\vec{R}]} e^{-\tilde{\mathcal{S}}[\vec{R}]}}{\int \mathcal{D}[R] e^{-\tilde{\mathcal{S}}[\vec{R}]}} = \frac{V_c^2}{U_a^2} \frac{\langle \mathcal{O} \rangle}{\langle \mathcal{O}_a \rangle} , \quad (30)$$

where  $U_a$  is the strength of the real reference potential. The advantage of this method is obvious from Eq. (30): one does not calculate the whole problem from the beginning, but uses a solved one as a reference problem. The computational effort is spent on calculating just the difference between both.



### 3 Random-Walk-Algorithms

The most direct implementation of the path integral expression for the scattering amplitude is the method of random walks. We will first discuss this approach for real potentials and later apply it to complex potentials. The procedure for generating paths starts with a set of initial points  $\vec{r}_1$ . From each of these points the next point is then reached by means of the Euclidean propagator  $\langle \vec{r}_i | e^{-\tilde{\mathcal{S}}} | \vec{r}_1 \rangle$ . The set of paths generated in this way is then distributed according to  $P_S$ . This procedure is continued until a sufficiently large time  $\beta$  is reached to allow the projector  $e^{-\beta\hat{H}}$  to filter out the zero energy contribution. As the projection time grows, more paths outside the potential region contribute, while configurations inside, which sample the dynamics of the target, only yield a small contribution [19]. This causes large fluctuations in the results and we examine methods to improve the generation of paths through the guided random walk (GRW) algorithms.

#### 3.1 Simple methods: free diffusion and diffusion with sources and sinks

*Free diffusion:* The simplest method to generate paths uses only the free action  $\mathcal{S}_0$  in the Euclidean propagator,

$$\mathcal{S}_0[\vec{R}] = \sum_{n=1}^N \frac{\mu}{2\varepsilon} (\vec{r}_{n+1} - \vec{r}_n)^2. \quad (31)$$

The matrix element in Eq. (23) is then evaluated by using paths distributed according to

$$P_S[\vec{R}] = \frac{1}{\mathcal{N}_\varepsilon^N \mathcal{N}_v} \exp \left[ -\mathcal{S}_0[\vec{R}] + \ln v_1 \right], \quad (32)$$

$$\mathcal{N}_v = \int d\vec{r} v(\vec{r}) < \infty. \quad (33)$$

The real potential  $U$  only enters through the generalized observable  $\mathcal{O}[\vec{R}]$ ,

$$\mathcal{O}[\vec{R}] = 2\pi \mu^2 U_0^2 \frac{\mathcal{N}_v}{\mathcal{N}_\beta} \exp \left[ \ln v_{N+1} - \mathcal{S}_U[\vec{R}] \right], \quad (34)$$

$$\mathcal{S}_U[\vec{R}] = \varepsilon \sum_{n=1}^{N+1} U_n. \quad (35)$$

The procedure to generate paths according to the distribution  $P_S[\vec{R}]$  consists of the following steps:

1. Choose  $Z_{in}$  starting coordinates  $\vec{r}_1^{(\alpha)}$ ,  $\alpha = 1, \dots, Z_{in}$  according to the initial distribution  $\Phi_v(\vec{r}_1)$ ,

$$\Phi_v(\vec{r}_1) = \frac{1}{\mathcal{N}_v} v(\vec{r}_1) . \quad (36)$$

We only discuss here the case that  $v(\vec{r})$  does not change sign, which is true for our choice of a Gaussian shape.

2. Each of the  $Z_{in}$  points is successively propagated with the transition probability  $p$ ,

$$p(\vec{r}_n, \vec{r}_{n+1}; \epsilon) = \frac{1}{\mathcal{N}_\epsilon} \exp \left[ -\frac{m}{2\epsilon} (\vec{r}_n^{(\alpha)} - \vec{r}_{n+1}^{(\alpha)})^2 \right] , \quad (37)$$

for the step from point  $n$  to  $n+1$  of a given path  $\alpha$ . This is numerically implemented by taking

$$\vec{r}_{n+1}^{(\alpha)} = \vec{r}_n^{(\alpha)} + \sqrt{\frac{\epsilon}{\mu}} \vec{\eta}_n^{(\alpha)} , \quad \alpha = 1, 2, \dots, Z_{in} , \quad (38)$$

where  $\vec{\eta}$  is a vector consisting of three independent, uniformly distributed random numbers.

3. Step 2 is repeated  $N = \frac{\beta}{\epsilon}$  times.
4. Finally, the scattering amplitude is obtained as

$$f_0^2 = \frac{1}{Z_{in}} \sum_{\alpha=1}^{Z_{in}} \mathcal{O}[\vec{R}^{(\alpha)}] . \quad (39)$$

*Replication of paths:* Properties of the potential can be incorporated into the generation of paths by the method of sources and sinks for paths: each of the segments of a path is assigned a weight  $D_n^{(\alpha)}$ ,

$$D_n^{(\alpha)} \equiv \exp \left[ -\epsilon U(\vec{r}_n^{(\alpha)}) \right] . \quad (40)$$

After each free diffusion step as described above, the segments of the path generated up to this 'time'  $n\epsilon$  are replicated or deleted, depending on the value of  $D^{(\alpha)}$ :

Let  $D_{int}$  be the largest integer which satisfies  $0 < D_{int} \leq D_n^{(\alpha)}$  and draw a random number  $w$  in the interval  $[0, 1]$ . For  $w \leq (D_n^{(\alpha)} - D_{int})$  one makes  $D_{int}$  copies of the path segment up to this point, otherwise only  $(D_{int} - 1)$  copies. These additional paths are then continued independently. If however  $D_{int} = 0$ , the segment will be kept with probability  $w$  or deleted with the complementary probability  $(1 - w)$ .

It is easily seen that this leads to the replication of paths in the region of attractive potentials; for repulsive potentials there is no replication of paths and the method ensures that paths with low weight factor  $D^{(\alpha)}$  are not followed too far. Especially for the case of strongly attractive potentials, not all paths one obtains in this fashion are uncorellated, since the paths from replication contain common segments. Compared to the free diffusion, the advantage of this method is that the paths are distributed according to  $P_S$ , where  $S$  is the full action, *i.e.* they are selected in a way that reflects the dynamics of the system.

We now illustrate the above by applying these simple random walk methods to a repulsive and an attractive real potential. To be able to judge the quality of the methods, we calculated the exact expectation value of the observable as a function of the Euclidean time  $\beta$ . This was done by solving Eq. (13) in its phase space representation with the fast Fourier transform (FFT) method [20]. Figs. 1 and 2 show results obtained with the free diffusion and the path replication method. By comparing to the FFT calculation, we see that for an attractive potential the convergence is much slower and relatively large projection times, typically  $\beta = 100fm$ , are needed. This long projection time requires high stability of the calculations. In the following we will therefore often use the attractive potential as a test case.

The difference in quality between the two methods, free diffusion and path replication, is small. The main source of this insensitivity can be traced to the rapid diffusion from the potential region, which leads to a relatively small number of paths that actually contributes to the matrix element of the observable: starting with 100000 paths, the number grows to 135000 for the attractive potential. Of these, only 630 contribute at  $\beta = 100fm$ , *i.e.* the generation of new paths does only little to balance the loss of paths due to the free diffusion from the potential region. For a repulsive potential, the number of paths drops from the initial 100000 to 50000, of which only 120 contribute. For free diffusion, the total number of paths stays constant and 330 contribute. The number of contributing paths is not sufficiently different in the cases considered to lead to a noticeable difference in accuracy. In view of the achieved accuracy and the long projection time, none of the methods seems suitable for application to complex observables and many-body targets, which have higher demands on numerical accuracy and speed of the calculations.

### 3.2 Guided random walk algorithms

An improvement of the random walk method can be achieved if an approximate solution for the wavefunction of the system is known. This solution can be used for the generation of the paths one uses to sample a modified observable. For the scattering from a many-body target such an approximate solution can be obtained for example from an effective single particle potential [3]. Of course, since we consider the potential scattering of a particle, the exact wavefunction in our present studies can be obtained by solving the Schrödinger equation with standard numerical methods. To get an impression of the power of the guided random walk method, we use here the 'optimal' situation: we work with the exact wavefunction for the real potential. This provides an upper limit for the performance of this method, since with only an approximate wavefunction the accuracy can be expected to get worse.

For the zero energy case, we thus use the solution of the Hamiltonian with a real potential,  $U(\vec{r}) = U_0 v(\vec{r})$ , that satisfies

$$H \xi(\vec{r}) = 0. \quad (41)$$

The 'time-independent guided random walk method' only makes use of this stationary solution. We have tested the time independent method to generate paths and evaluate the appropriate observable. It was found that in our case this method also could not prevent the substantial drifting of paths out of the potential region. Consequently, there was no noticeable improvement in accuracy compared to the simple random walk method. We therefore describe here only the time dependent guided random walk, which tries to compensate the loss of configurations. An essentially constant density of configurations in the potential region can be ensured by using a zero energy wavefunction with a norm that depends on the Euclidean time,  $\tau_e$ ,

$$\phi(\vec{r}, \tau_e) = \tau_e^{\frac{3}{2}} \xi(\vec{r}). \quad (42)$$

One then considers the evolution of the product  $\phi(\vec{r}, \tau_e) \psi(\vec{r}, \tau_e)$ , which has the infinitesimal evolution operator

$$\hat{\rho}_\varepsilon = \phi(\hat{r}, \tau_e + \varepsilon) e^{-\varepsilon \hat{H}} \phi^{-1}(\hat{r}, \tau_e). \quad (43)$$

It can be shown [21] that the matrix element can be written in terms of integrals over paths distributed according to a modified distribution

$$P_S[\vec{R}] = \frac{\beta_2^{\frac{3}{2}}}{\mathcal{N}_v} \prod_{n=1}^N \left\{ \frac{1}{\mathcal{N}_{\varepsilon_n}} \exp \left[ -\frac{\mu}{2 \varepsilon_n} (\vec{r}_{n+1} - \vec{r}_n - \frac{\varepsilon}{\mu} \vec{F}_n)^2 \right] \right\} v(\vec{r}_0), \quad (44)$$

$$\mathcal{N}_{\varepsilon_n} = \left[ \frac{2\pi\varepsilon_n}{\mu} \right]^{\frac{3}{2}}, \quad \vec{F}_n = \vec{\nabla} \ln \xi_n, \quad \varepsilon_n = \varepsilon \left[ 1 + \frac{\varepsilon}{\mu} \Delta \ln \phi_n \right], \quad (45)$$

and a simple modified observable,

$$\mathcal{O}[\vec{R}] = \mu (2\pi\mu)^{-\frac{1}{2}} U_0^2 \mathcal{N}_v \exp [\ln v_{N+1} - \ln \xi_{N+1} + \ln \xi_1]. \quad (46)$$

After discretization, the infinitesimal evolution operator reads

$$\rho_\varepsilon = \left[ \frac{n+1}{n} \right]^{\frac{3}{2}} \frac{1}{\mathcal{N}_{\varepsilon_n}} \exp \left[ -\frac{\mu}{2\varepsilon_n} (\vec{r}_{n+1} - \vec{r}_n - \frac{\varepsilon}{\mu} \vec{F}_n)^2 \right]. \quad (47)$$

The numerical implementation of the above is similiar to the diffusion case: successive points of a path are obtained according to

$$\vec{r}_{n+1}^{(\alpha)} = \vec{r}_n^{(\alpha)} - \frac{\varepsilon}{\mu} \vec{F}_n^{(\alpha)} + \sqrt{\frac{\varepsilon_n}{\mu}} \vec{\eta}_n^{(\alpha)}, \quad \text{where } \alpha = 1, 2, \dots, Z_{in}. \quad (48)$$

After each step, there is a replication of the path segment by a factor

$$D_n \equiv \left[ \frac{n+1}{n} \right]^{\frac{3}{2}}, \quad (49)$$

through a procedure analogous to the one in Section 3.1. The difference is that now the weight factor is not space-, but time-dependent, reflecting the time-dependent norm of  $\phi$ . This change of the particle number was found to be successful in keeping the number of points in the potential region at a reasonably high level. The result is an increase in accuracy and stability, as shown in Fig. 3.

The price one pays is that the total number of points grows rapidly with time. To keep about 2000 points in the potential region over a projection time of  $\beta = 100fm$ , an initial number of 100 configurations grows to ca. 200000. Since the generated paths all go back to a very small number of initial configurations, there is a noticeable correlation among the paths for small  $\beta$ , which can be seen by the large initial deviations from the exact result. However, for large projection times the method is remarkably stable. This is also due to the presence of the drift-force  $\vec{F}$  in Eq. (48), which together with the random configurations  $\eta$  has a decorrelating effect. Such effects will be discussed in more detail in the next chapter in connection with the Langevin algorithm.

Due to the reasonable stability, the time-dependent GRW method looks like a promising method for the complex potentials, which require higher numerical accuracy. The full Hamilton-operator  $\hat{H}$  is now

$$\hat{H} = \hat{K} + \hat{U} + i\hat{W}. \quad (50)$$

The wavefunction  $\phi(\vec{r}, \tau_e)$  is determined through the real potential with the time-independent part satisfying

$$[K + U(\vec{r})] \xi(\vec{r}) = 0. \quad (51)$$

The modified observable also includes also the absorptive part of the potential through the imaginary action  $\mathcal{S}_W[\vec{R}]$ ,

$$\mathcal{O}[\vec{R}] = \mu (2\pi\mu)^{-\frac{1}{2}} U_0^2 \mathcal{N}_v \exp \left[ \ln v_{N+1} - \ln \xi_{N+1} - \mathcal{S}_W[\vec{R}] + \ln \xi_1 \right]. \quad (52)$$

The resulting scattering length is shown in Fig. 4. The time dependent GRW does somewhat better than the random walk method based on the free action. The figure shows that there are less fluctuations in the results obtained with the time-dependent GRW. However, some fluctuations remain. They are mainly due to the phase, which sums up the contributions along each path. Similar results were also found for an absorptive potential with a repulsive real part and are not shown.

In summary, we have seen that the random walk algorithm converged reasonably fast towards the exact scattering length. However, the numerical accuracy was not satisfactory for any of the different versions we tried. A typical calculation took ca. 25 minutes on a HP UX 9000/720. Thus while these algorithms are not useful for the calculation of scattering from a many-body target, they can provide estimates for the typical projection times one needs. These projection times are needed as an input in the algorithms described in the next chapters, where paths are not built up successively from point to point, but entire paths of a given length  $\beta$  are generated and modified.

## 4 Langevin and Hybrid Algorithms

### 4.1 Standard Langevin Algorithm

The path integral methods presented in the last section describe a diffusion process in the physical time  $\beta$ , sequentially generating the elements of a path,  $R_i$ . Another approach is the Langevin algorithm, which updates an entire given path  $\vec{R}$ . This development of a path proceeds in terms of a new variable  $\tau$ , the 'Langevin time', and takes place according to the stochastic differential Langevin equation,

$$\frac{\partial}{\partial \tau} R_i(\tau) = -\tilde{S}_i'(\tau) + \pi_i(\tau) , \quad (53)$$

where  $\pi$  is a Gaussian random variable

$$\langle \pi_i(\tau) \rangle = 0 , \quad \langle \pi_i(\tau), \pi_j(\tau') \rangle = 2 \delta_{ij} \delta(\tau - \tau') . \quad (54)$$

The variable  $\tau$  is different from the physical time, which is represented here by the index  $i$ . When discretized,  $\tau$  labels the elements in a Markov chain. Again,  $\tilde{\mathcal{S}}$  is a real approximation to the action  $\tilde{\mathcal{A}}$  and

$$\tilde{S}_i'(\tau) = \frac{\partial}{\partial R_i} \tilde{\mathcal{S}}[\vec{R}(\tau)] . \quad (55)$$

The stochastic process described by Eq. (53) is ergodic,

$$\langle \mathcal{O} \rangle_P \equiv \int \mathcal{D}[\vec{R}] P_S[\vec{R}] \mathcal{O}[\vec{R}] = \lim_{\tau \rightarrow \infty} \frac{1}{\tau} \int_0^\tau d\tau' \mathcal{O}[\vec{R}(\tau')] , \quad (56)$$

where

$$P_S[\vec{R}] = \frac{1}{\mathcal{N}_S} \exp \left[ -\tilde{\mathcal{S}}[\vec{R}] \right] . \quad (57)$$

Choosing for  $\tilde{\mathcal{S}}$  the real part of the action, the paths generated through Eq. (53) will be used to evaluate the modified observable

$$\mathcal{O}[\vec{R}] = \exp \left[ -\mathcal{S}_W[\vec{R}] \right] . \quad (58)$$

For numerical evaluation, Eq. (53) is discretized. We first write the contributions to the action  $\tilde{\mathcal{S}}$ ,

$$\tilde{\mathcal{S}}[\vec{R}] = \tilde{\mathcal{S}}_0[\vec{R}] + \mathcal{S}_U[\vec{R}] , \quad (59)$$

in a convenient matrix notation

$$\tilde{\mathcal{S}}_0[\vec{R}] = \mathcal{S}_0[\vec{R}] + \frac{1}{2b^2} (\vec{r}_1^2 + \vec{r}_{N+1}^2) \equiv \vec{R} \mathbf{A}_0 \vec{R}, \quad (60)$$

$$\mathcal{S}_U[\vec{R}] = \varepsilon \sum_{i=1}^{N+1} U_i \equiv \frac{\varepsilon}{3} \text{Tr } \mathbf{D}_U[\vec{R}], \quad (61)$$

$$U_i \equiv \begin{cases} \frac{1}{2} U v(\vec{r}_i) & \text{for } i = 1, N+1 \\ U v(\vec{r}_i) & \text{for } i = 2, 3, \dots, N \end{cases}. \quad (62)$$

The matrix  $\mathbf{A}_0$  is symmetric and has  $3(N+1)$  eigenvalues,  $\lambda_i > 0$ , while  $\mathbf{D}_U[\vec{R}]$  is diagonal and built from  $N$  blocks,

$$\mathbf{D}_U[\vec{R}] = [U_i \mathbf{E}_3] , \quad i = 1, 2, \dots, (N+1), \quad (63)$$

where  $\mathbf{E}_3$  denotes the three-dimensional unit matrix. With these definitions we obtain the discretized version of the Langevin equation that updates a path  $\vec{R}^{(n)}$  and yields  $\vec{R}^{(n+1)}$ ,

$$\vec{R}^{(n+1)} = [\mathbf{1} - \delta \mathbf{M}^{(n)}] \vec{R}^{(n)} + h \vec{\pi}^{(n)}, \quad (64)$$

$$\mathbf{M}^{(n)} = \mathbf{A}_0 - \frac{\varepsilon}{b^2} \mathbf{D}_U^{(n)}. \quad (65)$$

The continuous variable  $\tau$  in the Langevin equation has been replaced by the discrete label  $n$ ,

$$n = \frac{\tau}{\delta}, \quad (66)$$

where  $\delta$  is the stepsize and we have defined  $h$  through

$$\delta \equiv \frac{h^2}{2}. \quad (67)$$

The Langevin algorithm thus generates from a starting path  $\vec{R}^{(1)}$  a sufficiently large number of path updates through Eq. (64), which are used to evaluate the observable. However, not all successive paths produced by the iteration of Eq. (64) are independent and to assess the efficiency of the algorithm we have to consider the autocorrelation time  $\tau_{cor}$  separating two decorrelated paths. As can be seen from Eq. (64), the evolution in  $\tau$  is governed by the eigenvalues  $\mu_i$  of the matrix  $\mathbf{M}$ . It can be shown [25] that

$$\tau_{cor} \sim \frac{1}{\mu_{min}}, \quad (68)$$



where  $\mu_{min}$  is the smallest eigenvalue of  $\mathbf{M}$ . On the other hand, for the algorithm to be numerically stable, the step size  $\delta$  should be small enough to resolve also the evolution of the quickly changing modes, *i.e.*

$$\delta \ll \frac{2}{\mu_{max}}. \quad (69)$$

Consequently, the number of intermediate steps  $n_{cor}$ , required such that a path  $\vec{R}^{(m)}$  is decorrelated from its predecessor  $\vec{R}^{(n)}$ , satisfies

$$n_{cor} = \frac{\tau_{cor}}{\delta} \gg \frac{\mu_{max}}{\mu_{min}}, \quad m = n + n_{cor}. \quad (70)$$

In general, the spectrum of the full matrix  $\mathbf{M}$  cannot be calculated. Since our scattering potential has a short range, the potential matrix  $\mathbf{D}_{\mathbf{U}}^{(n)}$  vanishes in most regions of the space of paths. As an approximation, we therefore replace the full dynamical matrix,  $\mathbf{M}$ , by the matrix containing the free action,  $\mathbf{A}_0$ , and get a lower limit for the autocorrelation time in terms of the eigenvalues  $\lambda_i$ ,

$$n_{cor} \gg \frac{\lambda_{max}}{\lambda_{min}}. \quad (71)$$

The ratio of the largest and smallest eigenvalues is shown in Fig. 5. As can be seen, a large number of intermediate steps must be taken before an uncorrelated path is obtained. This is due to the presence of two very different length scales in scattering: The average the distance of the middle of the path from the origin is of order  $O[\frac{\beta}{\mu}]$  and grows with the projection time. On the other hand, since the endpoints of the paths are required to lie in the potential, a small scale  $b$  enters. Of course, the actual spectrum of eigenvalues in scattering is continous and the presence of a lowest eigenvalue  $\lambda_{min} > 0$  is due to discretization. The finite projection time, the existence of a condition for the endpoints and the discretization of the action are thus crucial for the convergence of the Langevin algorithm in scattering problems [3].

The change of the stability criteria due to the potential can be roughly estimated in the region where  $\mathbf{D}_{\mathbf{U}}$  is approximately constant. For a path entirely in this region one has

$$\delta \ll \frac{2}{\lambda_i - \frac{\varepsilon U_0}{b^2}}, \quad (72)$$

For an attractive potential, numerical stability may therefore require a stepsize  $\delta$  which is considerably smaller than the estimate based on the eigenvalues of  $\mathbf{A}_0$ ;

for repulsive potentials the stepsize may be taken larger, at least as long as the values of  $U_0$  are such that the r.h.s. of Eq. (72) stays positive and finite and the condition doesn't become meaningless. The presence of the potential thus may require an increase in the number of intermediate steps one needs before another uncorrelated path is obtained or, even worse, it can lead to numerical instabilities. The condition Eq. (72) does not apply in the region where the potential varies significantly. Lowering the stepsize in the physical time,  $\varepsilon$ , to reduce the influence of the potential on the stability it is no remedy, since the eigenvalues  $\lambda_i$  also depend on  $\varepsilon$ . However, since the range of the potential is relatively small, most steps take place in the outer region and Eq. (68) still provides a reasonable indicator for the overall autocorrelation time.

Finally, to avoid local instabilities and errors due to the discretization of  $\tau$ , the updated path obtained from the discretized Langevin equations are treated as the proposal step in a Metropolis procedure [26]. The resulting algorithm is often referred as the 'exact' Langevin algorithm.

The above considerations are illustrated by the examples in Figs. 6 and 7. For a potential with a strong repulsive real part and strong absorption as in Fig. 6, a projection time of  $\beta = 50 \text{ fm}$  and  $\varepsilon = 0.5 \text{ fm}$  is sufficient; as in Section 2.1 this was checked by comparing to the results of deterministic FFT methods [20]. Inspection of Fig. 5 shows that one needs according to Eq. (71)  $n_{cor} \gg 5000$  intermediate Langevin steps in  $\tau$  to generate an independent path. The numerical values for  $\lambda_{max}$  and  $\lambda_{min}$  are 20 and  $4.7 \cdot 10^{-3}$ . The stepsize  $\delta$  is chosen to be  $2.5 \cdot 10^{-2}$  in order to meet condition Eq. (69) at least in the region where the potential vanishes. Therefore the result shown in Fig. 6 contains less than 80 decorrelated paths. After an initial equilibration phase of about 5000 Langevin steps, the fluctuations get smaller and the result for the real part of the scattering length gets very close to the exact value. However, the imaginary part is still quite far off and shows little sign of further convergence to the exact answer with increasing  $\tau$ .

For attractive potentials the situation is worse and the necessary projection time increases. For the strength parameters in Fig. 7,  $U_0 = -0.3 \text{ fm}^{-1}$  and  $W_0 = -1.0 \text{ fm}^{-1}$ , the FFT calculations show that a projection time  $\beta = 80 \text{ fm}$  should be chosen. This introduces a larger length scale and the number of intermediate steps  $n_{cor}$  increases quickly. For  $\varepsilon = 0.5 \text{ fm}$  the number  $n_{cor}$  is already of the order of 10000 and less than 40 independent paths contribute to the result in Fig. 7. Correspondingly, one has  $\lambda_{max} = 20$  and  $\lambda_{min} = 1.9 \cdot 10^{-3}$ ;  $\delta$  is chosen to be

$1.0 \cdot 10^{-2}$ . The equilibration time is now larger, ca. 10000 steps, and fluctuations in the result remain as  $\tau$  increases. One improvement would be to reduce discretization errors in  $\beta$  for the longer projection time by reducing the stepsize  $\varepsilon$ . This would lead to an even larger  $n_{cor}$  as illustrated by Fig. 5. However, the calculation in Fig. 7 already took approximately 2.5 hours on one processor of a CONVEX EXEMPLAR 1000 using standard tools for algebraic manipulations. Therefore this possibility of improving the performance seems not practical.

The rather poor performance of the Langevin algorithm seen in the two examples considered here is very different from the successful application to real potentials in Ref. [3], which could even be extended to many-body targets. The major cause of this difference is the phase in the modified observable, Eq. (58), which is responsible for strong fluctuations. Clearly the few independent paths in the examples above are not sufficient to probe the rapidly varying observable and their number has to be increased to achieve a satisfactory result. Straightforward extensions of the above method by increasing the number of Langevin steps would lead to prohibitively long computation times. We therefore now consider two possibilities which can help to accelerate the algorithm: The first one modifies the dynamical properties of the Langevin process by introducing a kernel into the stochastic differential equation, Eq. (53), while the second method, the hybrid algorithm, combines the Langevin algorithm with a microcanonical approach.

## 4.2 Langevin Algorithm with Kernel

To speed up the convergence of the Langevin algorithm, an appropriate matrix kernel,  $\mathbf{K}$ , can be inserted into the discretized Langevin equation, Eq. (64). This does not change the distribution  $P_S[\vec{R}]$  according to which the paths are sampled. The conditions under which the use of a kernel is valid are discussed in *e.g.* Refs. [22] - [24]. For position-independent kernels the equation becomes

$$\vec{R}^{(n+1)} = \left[ \mathbf{1} - \frac{h^2}{2} \mathbf{K} \left[ \mathbf{A}_0 - \frac{\varepsilon}{b^2} \mathbf{D}_U^{(n)} \right] \right] \vec{R}^{(n)} + h \sqrt{\mathbf{K}} \vec{\pi}^{(n)}. \quad (73)$$

Note that the square root of the matrix  $\mathbf{K}$  re-scales  $h$ , the stepsize in the 'Langevin time'  $\tau$ . The dynamical matrix,  $\mathbf{M}$ , which now governs the evolution of the algorithm in  $\tau$ , is

$$\mathbf{M}^{(n)} = \mathbf{K} \left[ \mathbf{A}_0 - \frac{\varepsilon}{b^2} \mathbf{D}_U^{(n)} \right], \quad (74)$$

with  $n = \frac{\tau}{\delta}$ .

Again the region outside the potential dominates the dynamical properties of the algorithm. To increase the efficiency, it is desirable to decouple the slow, long ranged modes of the free action  $\tilde{\mathcal{S}}_0$ , corresponding to the small eigenvalues of  $\mathbf{A}_0$ , from the fast modes. A natural choice for the kernel is  $\mathbf{K} = \mathbf{A}_0^{-1}$ , which leads to

$$\mathbf{M}^{(n)} = \mathbf{1} - \frac{\varepsilon}{b^2} [\mathbf{A}_0^{-1} \mathbf{D}\mathbf{U}^{(n)}], \quad (75)$$

reducing to the unity operator outside the potential region and thus has eigenvalues  $\lambda_i = 1$ . This lowers the correlation time, Eq. (68), quite dramatically to  $\tau_{cor} \sim 1$ !

As discussed in the previous section, numerical instabilities can occur in the region of the potential. With the choice  $\mathbf{K} = \mathbf{A}_0^{-1}$  the potential does not enter into the kernel and the scaling of the stepsize in the inner and the outer region of the potential is the same. Nevertheless, since the correlation time is dominated by the behavior in the outer region, where the different eigenmodes are decoupled by the use of the above kernel, far less intermediate steps are required to generate the necessary independent paths.

The improvement of the algorithm through the kernel does not permit an increase in the stepsize  $\delta$ . The coupling of the modes in the interior remains and the numerical stability requires again to follow each step proposed by the Langevin equation with a kernel by a Metropolis acceptance-step. Since the algorithm is dominated by the long ranged modes of the free action, a large rejection rate is obtained. This has to be remedied by a decrease in the step-size  $\delta$  compared to the case without kernel in order to have the Metropolis step accept 50% to 80% of the proposed paths. Another factor that leads to longer computation times is the matrix structure of the kernel and thus the increased algebraic complexity of the algorithm. The number of the required numerical operations scales at least quadratically with the dimension of the matrix  $\mathbf{A}_0$ .

Fig. 8 shows the result for a complex potential with a repulsive real part. The calculation with the projection time  $\beta = 80$  fm, consumed approximately 2 hours, the runs with  $\beta = 50$  fm and different discretization  $\varepsilon$  took 1 hour and 3 hours, respectively. The computational effort and the accuracy is similar as in the calculation without a kernel in Fig. 6. But the increased number of about 5000 independent paths leads to a very stable behaviour; furthermore the number of steps for the initial equilibration is now reduced to about 1000. As can be concluded from Fig. 8, a smaller discretization in the projection time  $\beta$  is needed to improve the accuracy of the result. This trend was verified by a run with  $\varepsilon = 0.1$  fm,  $\beta = 50$  fm and

$n = 5000$  Langevin steps, which however took more than 5 hours (not shown).

Fig. 9 shows results for a potential with an attractive real part and  $\beta = 80$  fm, which took about 2 hours. Although the results are again remarkable stable after relatively few Langevin steps due to the use of a kernel, they deviate from the exact results. The calculations were repeated (see Fig. 10) with a longer projection time,  $\beta = 200$  fm, and fewer Langevin steps,  $n = 5000$ , which required 5 hours of CPU time. The results are somewhat improved, but are less stable. The fluctuations are caused by the decreased number of independent paths. Calculations for more attractive potentials were also carried out and it was found that no reasonable convergence could be achieved within the chosen time limits.

The usefulness of introducing a kernel into the Langevin algorithm is therefore limited to a small range of potential parameters, where it can produce very stable results. However, as this improved algorithm is rather time-consuming, its application to multi-particle systems looks not promising.

### 4.3 Hybrid-Algorithm

The Langevin algorithm can be combined with deterministic molecular-dynamical methods to a 'hybrid algorithm' with the advantages of both methods [8, 27, 28, 29, 30]. The deterministic updating of entire paths in a new 'hybrid time',  $t$ , is governed by a 'CPU Hamiltonian',  $\mathcal{H}$ , which is different from the physical Hamiltonian,  $H$ ,

$$\mathcal{H} = \frac{1}{2}\vec{\pi}^2 + \tilde{\mathcal{S}}[\vec{R}] \quad (76)$$

$$\frac{\partial}{\partial t} R_i = \frac{\partial \mathcal{H}}{\partial \pi_i} \quad (77)$$

$$\frac{\partial}{\partial t} \pi_i = -\frac{\partial \mathcal{H}}{\partial R_i} = -\tilde{\mathcal{S}}'_i[\vec{R}]. \quad (78)$$

In this approach  $\vec{\pi}$  is the CPU momentum of an object with position vector  $\vec{R}$ , moving in a potential  $\tilde{\mathcal{S}}[\vec{R}]$ . The discretized version of these equation for the transition from  $n$  to  $n+1$  reads to order  $0[h^2]$

$$T_2(h) \sim \begin{cases} \pi_i^{(n+\frac{1}{2})} &= \pi_i^{(n)} - \frac{1}{2} h \tilde{\mathcal{S}}'_i{}^{(n)}, \\ R_i^{(n+1)} &= R_i^{(n)} + h \pi_i^{(n+\frac{1}{2})}, \\ \pi_i^{(n+1)} &= \pi_i^{(n+\frac{1}{2})} - \frac{1}{2} h \tilde{\mathcal{S}}'_i{}^{(n+1)}. \end{cases} \quad (79)$$

After several of deterministic steps according to the above equations of motion, the momentum  $\vec{\pi}$  is updated by a Gaussian-distributed random number. Therefore, all

momenta and position vectors, if they belong to a deterministic step or not, are random variables.

The trajectory of a path  $\vec{R}$  is split into sub-trajectories,  $\mathcal{T}_\alpha$ , each starting with a Gaussian random momentum,  $\vec{\pi}^{(n_{I\alpha})}$ , and ending with  $\vec{\pi}^{(n_{F\alpha})}$ , containing

$$\Delta n_\alpha = n_{F\alpha} - n_{I\alpha}. \quad (80)$$

successive deterministic steps  $T_2(h)$ . The next sub-trajectory  $\mathcal{T}_{\alpha+1}$  then starts again with a Gaussian random momentum. Each move along a sub-trajectory is completed by a Metropolis acceptance step: The entire sub-trajectory is considered as a proposed move, which is accepted with probability

$$P_{acc} = \min \left[ 1, e^{-\Delta\mathcal{H}} \right], \quad (81)$$

$$\Delta\mathcal{H} = \mathcal{H}[\vec{R}^{(n_{F\alpha})}, \vec{\pi}^{(n_{F\alpha})}] - \mathcal{H}[\vec{R}^{(n_{I\alpha})}, \vec{\pi}^{(n_{I\alpha})}]. \quad (82)$$

This combination with the Metropolis step, which we have used in all examples below, is referred to as 'exact hybrid algorithm'.

The length of  $\mathcal{T}_\alpha$ , *i.e.* the number of deterministic steps  $\Delta n_\alpha$ , should be such that the paths  $\vec{R}^{(n_{I\alpha})}$  and  $\vec{R}^{(n_{F\alpha})}$  are decorrelated. Note that the first two steps of Eqs. (79) can be recombined to yield the ordinary Langevin equation, Eq. (64). In the Langevin case, a sub-trajectory  $\mathcal{T}$  then just consists of one step and the Metropolis step in both methods is analogous. The Hamiltonian  $\mathcal{H}$  describes a classical motion confined in a high-dimensional harmonic oscillator potential. In contrast to the previous approaches, the correlation between two paths both belonging to the same sub-trajectory has not an exponential but oscillatory dependence on the time which labels successive paths. The period of this dependence is  $t_i = \frac{1}{\sqrt{\lambda_i}}$ , where  $\lambda_i$  again denotes the eigenvalues of the free action matrix  $\mathbf{A}_0$ . To assure decorrelation of the smallest mode within one sub-trajectory, its length should satisfy

$$t_{tra} \sim \frac{1}{\sqrt{\lambda_{min}}}. \quad (83)$$

However, this does not prevent strong correlations between some other modes due to their different periodic time dependence of their correlation. Therefore in practice the length of each sub-trajectory is varied randomly in the vicinity of  $t_{tra}$ . Stability requires a number,  $n_{tra}$ , of steps in a subtrajectory that satisfies

$$n_{tra} = \frac{t_{tra}}{h} \sim \sqrt{\frac{\lambda_{max}}{\lambda_{min}}}. \quad (84)$$

Comparison of Eqs. (84) and (70) shows that the decorrelation in the hybrid algorithm is faster by a factor  $\sqrt{\frac{\lambda_{max}}{\lambda_{min}}}$  compared to the standard Langevin algorithms; the substantial size of this factor can be seen in Fig. 5. In addition to the decrease in the required number of intermediate steps, a further advantage of the hybrid algorithm is that it needs fewer Gaussian random numbers, which are very time consuming to generate.

Eqs. (79) represent only the simplest discretization in the time  $t$ . With increasing sub-trajectory length the discretization error increases also the drift from the CPU energy shell in the deterministic steps of a sub-trajectory. This drift is measured by  $\Delta\mathcal{H}$ , Eq. (82). Therefore the acceptance rate is lowered considerably and an improvement of the discretization in Eqs. (79) is necessary. We have studied examples with projection times in the range  $\beta = 50 - 200$  fm and stepsizes  $\varepsilon = 0.1 - 1.0$  fm. For sub-trajectories containing approximately  $n_{tra} = 50$  steps, the  $O[h^2]$  discretization was found adequate. For sub-trajectories of about 100 steps a discretization procedure up to terms of order  $O[h^4]$  and for 200 steps even  $O[h^6]$  was needed to lower  $\Delta\mathcal{H}$ , requiring three or nine times more steps, respectively, as with the  $O[h^2]$  discretization. We used the higher order scheme of Ref. [26], where Eq. (79) is extended to  $O[h^{k+2}]$  by the recursive relation

$$T_{k+2}(h) = T_k(\tilde{h}) T_k(-s\tilde{h}) T_k(\tilde{h}) \quad (85)$$

$$\tilde{h} = \frac{1}{2-s} h, \quad s = 2^{\frac{1}{k+1}}. \quad (86)$$

We found that in the range of parameters shown in Fig. 5 the hybrid algorithm, while technically more involved, works efficiently; discretization to order  $O[h^4]$  was sufficient.

The calculations shown in Figs. 11 and 12 are obtained with the exact hybrid algorithm, without a kernel. The potential is  $V_c = (2.3, -7.0) \text{ fm}^{-1}$ , the projection time  $\beta$  and the discretization  $\varepsilon$  are varied. As before, we found for this repulsive potential a projection time of  $\beta = 50$  fm sufficient. In this case discretization in the stepsize,  $h$ , to order  $O[h^4]$  was needed for  $\varepsilon = 0.33$  fm, while for  $\varepsilon = 0.1$  fm also terms of order  $O[h^6]$  were necessary. The calculations of discretization,  $\varepsilon = 0.33$  fm, and order,  $O[h^4]$ , consumed about 30 minutes of computing time, while the  $\varepsilon = 0.10$  fm,  $O[h^6]$  example needed more than 5 hours! Note that compared to  $O[h^4]$  calculations not only the number of algebraic manipulations is increased by a factor of three, but according to Eq. (84) and Fig. 5, longer projection times  $\beta$  and smaller values

of  $\varepsilon$  also lead to an significant increase in  $n_{tra}$ . The length of one trajectory can be estimated to range from  $t_{tra} \sim 15$  to  $t_{tra} \sim 30$  (see Fig. 5). This means, that Fig. 12 contains approximately 6000 independent samples for the discretization  $\varepsilon = 0.33 \text{ fm}$  and 2500 samples for  $\varepsilon = 0.10 \text{ fm}$ . The algorithm shows a good and stable performance. Reasonable results can be obtained for a smaller number of steps in the 'hybrid time'  $t$  which required about 10 minutes of computational time. Figs. 11 and 12 show that the discretization is important, but a slight discrepancy in the imaginary part still persists.

The limitations of the method show up for attractive potentials. While the hybrid algorithm was found to work equally well for potentials with weak attractive parts down to  $-0.3 \text{ fm}^{-1}$ , it starts to fail for more attractive values of  $U_0$ . This is due to the fact that with our standard parametrization of  $b = 0.5 \text{ fm}$  and  $\mu = 2.5 \text{ fm}^{-1}$ , a real potential develops a bound state at  $U_0 = -1.2 \text{ fm}^{-1}$ . Fig. 13 shows the results for  $V_c = (-0.5, -1.5) \text{ fm}^{-1}$ . The time needed for the calculations was 90 minutes for a projection time  $\beta = 100 \text{ fm}$ . Attempts with  $\beta = 200 \text{ fm}$  (not shown) were also found insufficient to filter the ground state well enough and therefore the algorithm cannot be used for attractive potentials of this strength.

Both extensions of the Langevin algorithm — use of a kernel and hybrid method — show a comparable behaviour. In our present applications of the algorithm, all versions comprise a sequence of steps given by Eq. (79); if a kernel used,  $h$  is scaled by the square root of  $\mathbf{K}$ . For stability reasons  $h$  was chosen to be approximately five times smaller then minimum required. The hybrid algorithm performs slightly faster because it needs far less Gaussian random numbers. For refined discretizations and long projection times, the two improved versions become equally time-consuming. It is hard to compare both extended schemes in general, since the calculation time depends on the algebraic structure of the chosen kernel, the implementation of the stochastic and deterministic differential equations and the discretization in  $h$  for the hybrid algorithm, which will vary from case to case. A kernel can also be introduced into the hybrid algorithm. We did not examine this possibility in detail, but tests showed that long trajectories cause stability problems. Therefore, one has to decrease the step size further and no gain in velocity is obtained.



## 5 Cluster-Algorithms

The hybrid and Langevin Algorithms are hindered by the fact that the coupling of two points of a path is restricted to nearest neighbor interactions. The physical potential is purely local and acts only within a small region with typical scale  $b$  of the potential width. Outside this region one is left with the free action. Hence, strongly oscillating paths are suppressed and long range correlations of order  $\beta$  dominate. Only the end point contributions containing the logarithm of the potential prevent the paths from drifting away to infinity. Hence, the displacement of a path on a large scale proceeds very slowly.

To improve the computational efficiency it is desirable to speed up these slow long-ranged modes without neglecting small fluctuations introduced by the short range of the potential. One would like a non-local or collective-mode updating based on information over a larger range of coordinates. We considered already one example of a such an algorithm in Section 4, where we introduced a kernel into the Langevin algorithm in order to deal with modes of the path motion and not just individual points when generating new paths. This approach suffered from the fact that we were not able to diagonalize the complete action. A coupling of the different modes through the potential remained.

A similiar situation often occurs for spin systems with nearest neighbor interactions. It is well known that large-scale modes of spin waves lead to sizeable slowing down. To remedy this, Swendsen and Young introduced a new type of collective algorithm for Potts models [4]. Additional auxiliary degrees of freedom are introduced into the original model: groups of correlated spins are treated as new degrees of freedom which are called 'clusters'. Changes are applied to these collective objects and not to single physical degrees of freedom, the individual spins, which leads to a much more efficient and faster way to deal with such systems. Subsequently many related applications have been developed for Ising models,  $O(N)$  spin models [5] and lattice gauge theories [31], commonly referred to as 'cluster algorithms'. There is considerable freedom in how to define a cluster. Some special aspect of the physical problem has to be used in each case to define the appropriate collective variables.

In this section, we extend the idea of clusters from bound systems to scattering problems. Adjacent coordinate points of a path having similiar positions will under certain conditions be considered as clusters and dealt with collectively. The size of these clusters must reflect the two different length scales of the physical system

under consideration, the short range of the potential and the large exterior region. There are two basic steps of the cluster algorithm which are carried out to a given path  $\vec{R}$  to generate a new path,  $\vec{R}'$ . They are repeated until satisfactory convergence of the matrix element of the chosen observable is achieved:

1. The first step consists of grouping the points of a path into clusters. As will be discussed in detail below, clusters are designated in a way that will favor a decrease in kinetic energy when updating the path by moving these clusters in step 2.
2. Once the clusters have been defined, a number of them are reflected on a randomly oriented plane  $\mathcal{E}$ , yielding a new path  $\vec{R}'$ . This new path consists thus of some clusters which were reflected and some which did not change. The decision whether a cluster is reflected or not is chosen in a way which favors the decrease of the potential energy. To preserve ergodicity, the position of  $\mathcal{E}$  is changed each time this step is carried out.

Both steps together lead to a splitting into smaller clusters in regions where the scale of the potential becomes important and to larger clusters where the influence of the kinetic part of the action is dominant. This is a great advantage compared to *e.g.* the methods in Section 3. There are two differences between our method and the application of cluster methods to spins of fixed length. First, it is in our case not sufficient to reflect the coordinate points through a plane which contains the origin; instead the position of the plane is varied throughout the entire space. Second, the scattering potential influences the reflection probability of clusters and therefore scales the effective size of reflected clusters.

## 5.1 Introduction of auxiliary variables and collective updating of paths

To group points of a given path  $\vec{R}$  into clusters, first auxiliary 'bond' variables  $b_i$  are defined for each pair of adjacent points  $\vec{r}_i$  and  $\vec{r}_{i-1}$ . The  $b_i$  can take on the values 0 or 1. The whole set of these variables for a path is denoted by  $B$ . A cluster  $\mathcal{C}$  is a set of neighbouring coordinates  $\vec{r}_i$  which are connected by non-zero bonds  $b_i$

$$\mathcal{C}^s \equiv [\vec{r}_i, \vec{r}_{i+1}, \dots, \vec{r}_j], \quad (87)$$

where  $s$  stands for the set of indices  $i, i+1, \dots, j$  of coordinate points contained in  $\mathcal{C}^s$ .

To decide which value should be assigned to a 'bond' variable  $b_i$ , we use a plane  $\mathcal{E}$ , characterized by a normal vector  $\hat{n}$  and its distance  $\xi$  to the origin. We then consider the point  $\vec{r}_i^*$ , which is obtained by reflecting  $\vec{r}_i$  through the plane  $\mathcal{E}$ , *i.e.*  $\vec{r}_i^* \equiv \vec{r}_i^*[\vec{r}_i, \mathcal{E}]$ . The probability of setting the bond variable  $b_i$  to zero is then given by

$$P_{nb,i} = \min \left[ 1, \exp \left( -\Delta \mathcal{S}_0[\vec{\Delta}_i] \right) \right], \quad (88)$$

where  $\Delta \mathcal{S}_0$  is determined by the free part of the action,

$$\Delta \mathcal{S}_0[\vec{\Delta}_i] = \frac{\mu}{2\varepsilon} \left[ \vec{\Delta}_i^{*2} - \vec{\Delta}_i^2 \right], \quad (89)$$

$$\vec{\Delta}_i = \vec{r}_i - \vec{r}_{i-1}, \quad (90)$$

$$\vec{\Delta}_i^* = \vec{r}_i^* - \vec{r}_{i-1}. \quad (91)$$

$P[B \mid \vec{R}, \mathcal{E}]$  is then conditional probability to obtain a set of bonds  $B$  for given plane  $\mathcal{E}$  [5], [32],

$$P[B \mid \vec{R}, \mathcal{E}] = \prod_{i=2}^{N+1} [P_{nb,i} \delta_{b_i,0} + (1 - P_{nb,i}) \delta_{b_i,1}], \quad (92)$$

$$B = [b_2, b_3, \dots, b_{N+1}] \quad \vec{R} = [\vec{r}_1, \vec{r}_2, \dots, \vec{r}_{N+1}]. \quad (93)$$

Taking into account these additional variables, the probability distribution  $P_S[\vec{R}]$  can be extended to the combined distribution  $\mathcal{P}$

$$\mathcal{P}[\vec{R}, \mathcal{E}, B] = P[B \mid \vec{R}, \mathcal{E}] P_S[\vec{R}] P_{\mathcal{E}}[\hat{n}, \xi]. \quad (94)$$

Here  $P_S[\vec{R}]$  is as defined before in Eq. (57) and  $P_{\mathcal{E}}[\hat{n}, \xi]$  describes the probability of finding a plane  $\mathcal{E}$  with normal vector  $\hat{n}$  and at a distance  $\xi$  to the origin. Its form will be specified below; in general it does not depend on the path  $\vec{R}$ . One has by construction

$$P_S[\vec{R}] = \int d\xi \int d\Omega \sum_{\{b\}} \mathcal{P}[\vec{R}, \mathcal{E}, B], \quad (95)$$

where integration over  $d\Omega$  extends over all possible orientations of  $\hat{n}$ . In terms of the extended probability distribution,  $\mathcal{P}[\vec{R}, \mathcal{E}, B]$ , we can re-write the expectation value of the modified observable  $\mathcal{O}[\vec{R}]$  as

$$\langle \mathcal{O} \rangle = \int \mathcal{D}[\vec{R}] \mathcal{O}[\vec{R}] P_S[\vec{R}] = \int \mathcal{D}[\vec{R}] \int d\xi \int d\Omega \sum_{\{b\}} \mathcal{O}[\vec{R}] \mathcal{P}[\vec{R}, \mathcal{E}, B]. \quad (96)$$

We now have to specify how we update the path  $\vec{R}$  by reflecting entire clusters on the plane  $\mathcal{E}$ . We denote by  $\mathcal{C}^{s*}$  the cluster one obtains by reflecting  $\mathcal{C}^s$  on  $\mathcal{E}$ . The probability for a cluster to be reflected is given by

$$T[\mathcal{C}^{s*} | \mathcal{C}^s] = \alpha \min [1, \exp (-\Delta \mathcal{S}_U[\mathcal{C}^c])] , \quad (97)$$

$$\Delta \mathcal{S}_U[\mathcal{C}^c] = \mathcal{S}_U[\mathcal{C}^{c*}] - \mathcal{S}_U[\mathcal{C}^c]. \quad (98)$$

Here  $0 < \alpha < 1$  is a real parameter; its choice will be discussed below. The quantity  $\mathcal{S}_U[\mathcal{C}^s]$  is the contribution of the potential to the action for the points within the cluster  $\mathcal{C}^s$ . Clusters which contain the endpoints also contribute  $\ln v$  according to Eq. (16). As can be seen from the above prescription, the probability for reflecting a cluster is chosen such that a transition to clusters with a lower potential energy is favored. Since the new path  $\vec{R}'$  in general contains reflected clusters,  $\mathcal{C}^{s*}$ , as well as unreflected ones,  $\mathcal{C}^s$ , the total transition probability for given  $\mathcal{E}$  and  $B$  reads

$$\mathcal{T}_{\mathcal{E}B}[\vec{R}' | \vec{R}] = \prod_{s=1}^{N_r} T[\mathcal{C}^{s*} | \mathcal{C}^s] \prod_{n=1}^{N_{nr}} \{1 - T[\mathcal{C}^{n*} | \mathcal{C}^n]\} . \quad (99)$$

The first factor in the above equation contains only the  $N_r$  clusters which are reflected to transform  $\vec{R}$  into  $\vec{R}'$ , while the second is due to the  $N_{nr}$  clusters which retain their position. After having updated a path  $\vec{R}$  to  $\vec{R}'$ , the procedure is then repeated with now  $\vec{R}'$  as starting path to obtain a new  $\vec{R}'$  *etc.*, until a sufficient number of paths is obtained for the evaluation of the observable.

It remains to be shown that the transition from  $\vec{R}$  to  $\vec{R}'$  satisfies the criterion of detailed balance,

$$\frac{\mathcal{P}[\vec{R}', \mathcal{E}, B]}{\mathcal{P}[\vec{R}, \mathcal{E}, B]} = \frac{\mathcal{T}_{\mathcal{E}B}[\vec{R}' | \vec{R}]}{\mathcal{T}_{\mathcal{E}B}[\vec{R} | \vec{R}']} . \quad (100)$$

Making use of the general relation

$$\frac{\min[1, e^{-\kappa}]}{\min[1, e^{\kappa}]} = e^{-\kappa} , \quad (101)$$

one finds for the r.h.s. of Eq. (100)

$$\frac{\mathcal{T}_{\mathcal{E}B}[\vec{R}' | \vec{R}]}{\mathcal{T}_{\mathcal{E}B}[\vec{R} | \vec{R}']} = \exp \left[ - \sum_{s=1}^{N_r} \Delta \mathcal{S}_U[\mathcal{C}^s] \right] . \quad (102)$$

The sum runs only over those  $N_r$  clusters which are reflected to make the transition from  $\vec{R}$  to  $\vec{R}'$ . The l.h.s. of Eq. (100) can be re-written by making use of Eq. (101)

to yield

$$\frac{\mathcal{P}[\vec{R}', \mathcal{E}, B]}{\mathcal{P}[\vec{R}, \mathcal{E}, B]} = \exp \left[ -\Delta\mathcal{S}[\vec{R}', \vec{R}] \right] \frac{P[B | \vec{R}', \mathcal{E}]}{P[B | \vec{R}, \mathcal{E}]}, \quad (103)$$

where the exponent is given by

$$\Delta\mathcal{S}[\vec{R}', \vec{R}] = \tilde{\mathcal{S}}[\vec{R}'] - \tilde{\mathcal{S}}[\vec{R}] = \sum_{t=1}^{N_{tr}} \Delta\mathcal{S}_0[\vec{\Delta}_t] + \sum_{s=1}^{N_r} \Delta\mathcal{S}_U[C^s]. \quad (104)$$

The index  $t$  in Eq. (104) runs only over those  $N_{tr}$  junctions  $\vec{\Delta}_t$  which are changed into  $\vec{\Delta}_t^*$ ; junctions which are contained in one cluster and junctions between two clusters which are reflected simultaneously do not contribute. The remaining term in Eq. (103) is obtained from Eqs. (88), (92), which yield

$$\frac{P[B | \vec{R}, \mathcal{E}]}{P[B | \vec{R}', \mathcal{E}]} = \exp \left[ -\sum_{t=1}^{N_{tr}} \Delta\mathcal{S}_0[\vec{\Delta}_t] \right]. \quad (105)$$

Combining Eqs. (103), (104) and (105) and inserting the result into the l.h.s. in Eq. (100), it is readily seen that the detailed balance condition is satisfied.

## 5.2 Scaling of the Cluster Size and Choice of Parameters

Important for the convergence and accuracy of the algorithm are the size and relative distribution of the clusters. Too many large clusters leads to inefficient probing of the potential region, while too many small clusters slow down the algorithm. The cluster distribution depends on the function  $P_{\mathcal{E}}$  and the coefficient  $\alpha$  in Eq. (97). We will discuss these two aspects in this section.

Geometrical considerations show that two neighbouring points belong to different clusters if the line connecting these points is intersected by the plane  $\mathcal{E}$ . Furthermore, it is readily seen that the size of the path segment contained in a cluster and the relative distribution of clusters with different sizes depend on the average distance of the plane  $\mathcal{E}$  to the region of classical paths where there are many points of a path. A distribution  $P_{\mathcal{E}}$ , which mainly places the plane into these physically preferred areas, results in many intersections and thus a large number of small clusters. If on the other hand the plane  $\mathcal{E}$  is placed at large distances from these regions, it is likely to intersect the path only at a few points, resulting in relatively few, large clusters. Thus, while at first sight the setting of bonds and grouping of points into clusters seems to entirely depend on the kinetic energy through the condition in Eq.

(88), the above considerations show that also the potential enters into the eventual distribution of clusters through the action  $\tilde{\mathcal{S}}$ : If the algorithm is equilibrated and starts to converge, the paths will be denser in regions of local minima of the real action  $\tilde{\mathcal{S}}$ ; reflection planes placed into these regions will lead to a splitting into many small clusters that allow for an accurate probing of the details of the potential.

The location of the plane  $\mathcal{E}$  is important for another reason. If  $\mathcal{E}$  is placed between two classically allowed regions, which are separated by a repulsive potential well, the reflection makes the fast interchange of path segments between both minima possible. A very large number of intermediate steps would be needed if, instead of clusters, we dealt with individual points as in the other algorithms. With the Langevin method, for example, the path first has to get through the repulsive potential barrier step by step before it reaches the neighboring minimum. This advantage of the cluster method is thus especially advantageous for scattering problems with a more complicated target structure such as many-body targets. Projectile paths will not remain trapped in regions separated by potential barriers, but move quickly and probe efficiently the entire target region.

We now discuss the choice of the parameters defining the distribution  $P_{\mathcal{E}}[\hat{n}, \xi]$  of the reflection planes. The orientation of the plane  $\mathcal{E}$  is defined by the normal vector  $\hat{n}$ ; in spherical coordinates  $\hat{n} = (\cos \psi \sin \theta, \sin \psi \sin \theta, \cos \theta)$ . Since we are considering scattering problems for spherically symmetric potentials no direction in space should be preferred and therefore the variables  $\cos \theta$  and  $\psi$  are randomly distributed over the intervals

$$\cos \theta \in [-1, 1], \quad \psi \in [0, 2\pi]. \quad (106)$$

The distance of the plane to the origin,  $\xi$ , is taken to be of Gaussian type,

$$P[\xi] = \left[ \frac{1}{2\pi w^2} \right]^{\frac{1}{2}} \exp \left[ -\frac{1}{2w^2} \xi^2 \right]. \quad (107)$$

If the projection time is large enough such that it satisfies  $b^2 \ll \frac{\beta}{\mu}$ , it can be shown that one has for the mean square distance of the midpoint of the paths to the origin,

$$\langle d_M^2 \rangle \approx 3 \frac{\beta}{4\mu}, \quad (108)$$

while the mean square distance of the endpoints is

$$\langle d_{1,N+1}^2 \rangle \approx 3 b^2. \quad (109)$$

For repulsive potentials the choice  $w \sim b$  for the width of the distribution leads to the best results. By concentrating the position of the reflecting plane in this region around the origin, one avoids that large segments in the exterior region, typically located at a distance  $\sqrt{\langle d_M^2 \rangle} \gg b$  are reflected into the potential region. Avoiding these paths, which are practically all rejected, speeds up the algorithm and reduces fluctuations. At the same time, this positioning of the reflecting planes takes into account the detailed shape of the potential in updating a path. On the other hand, taking  $w$  too small would lead to too many almost symmetrical reflections, as the planes are concentrated close to the symmetry center of the potential. The updated paths then only differ in the kinetic energy, while the contribution from the potential remains essentially unchanged. As the observable we evaluate only depends on the absolute value of the distance of the points of a path to the origin, such updates would only increase the computational time for the algorithm. In the examples below with a repulsive real part of the potential, which were all carried out with a projection time of  $\beta = 80$  fm, we chose a width of  $w = 0.5$  fm.

Attractive potentials require a much longer projection time. As a result, the midpoint of the paths moves further away from the origin and a larger distribution parameter  $w \sim O[d_M/2]$  should be chosen, yielding planes  $\mathcal{E}$  located about halfway between the midpoint of a path and the origin. Exterior segments are then often reflected into the energetically favored potential region; the average time a cluster remains in the interior region is short, since the next path-update will tend to move it out of that region again. Symmetric reflections through the origin are suppressed with this choice for  $w$ . A disadvantage of this choice results for the endpoints of the path, which should remain in the potential region: a large  $w$  leads to reflections out of this region and thus to configurations with a low acceptance. We used *e.g.* for a projection time of  $\beta = 300$  fm a width of  $w = 2.0$  fm, while for  $\beta = 500$  fm a choice of  $w \approx 4.0$  fm yielded the best results.

The above discussion shows that one has to balance competing effects when choosing the distribution of the planes: If  $w$  is chosen too small the tendency exists that many clusters are reflected to a position at the same distance to the origin; a too large  $w$  leads to motions of the endpoints with a low acceptance. Both extremes significantly lower the convergence speed of the algorithm. Clearly a different shape of the surface on which one reflects is also possible to avoid these problems and to further improve the efficiency of the algorithm; we do not pursue this here.

The probability for the reflection of a cluster depends on the potential and on

the choice of the parameter  $\alpha$  in Eq. (97), which also has an important influence on the relative distribution of clusters of different sizes. The decision if a cluster should be reflected is made independently for each cluster. If we take  $\alpha \approx 1$ , Eq. (97) shows that a single cluster is then practically always reflected if the potential energy is not increased by this operation. Therefore in regions with no structure due to the potential, it is then likely that many adjacent clusters are reflected simultaneously and 'effective clusters' of large size are formed. Within the range of the potential small clusters are formed since the reflection probability, Eq. (97), now changes over short distances due to changes of the potential. The algorithm then provides a fast collective movement in regions where the potential vanishes and leads at the same time to a fine scaling when the path crosses the relatively small interaction region. Therefore the choice of  $\alpha \approx 1$  is especially suited for scattering problems. On the other hand for  $\alpha = 1$  a cluster would always be reflected and effective clusters would tend to stay together in the exterior. This leads to a slowing down due to insufficient decorrelation of modes with a longer range. We obtained the best results with  $\alpha \approx 0.75$ . In choosing the parameters to optimize the cluster motions in the exterior we do not have to worry about how this affects the modified observable. In our case the observable depends on the imaginary part of the potential and has no structure outside of the range of the potential. The choice of  $\alpha$  can thus entirely be used to speed up the algorithm in the large exterior region.

Fig. 14 demonstrates for a repulsive potential how the algorithm selects automatically the appropriate scale for the clusters. Small clusters, necessary to probe the potential region in detail, can be found at short distances from the origin, while larger clusters can be found at larger distances. That the cluster size does not grow stronger at very large distances from the origin is due to the fact that the endpoints of the path are restricted to lie in the potential region. Too large clusters lead to too few acceptable paths.

### 5.3 Examples

We first show in Fig. 15 results for a potential with a repulsive real part. The parameters are the same as in Fig. 11, where it was shown that with the hybrid method a discretization of the projection times  $\beta$  down to  $\varepsilon = 0.25$  fm could not achieve convergence to the correct answer in both real and imaginary part. The results were obtained with the cluster algorithm for a refined discretization of  $\varepsilon =$



0.1 fm. There is now a rapid convergence to the correct result for the scattering amplitude. For the hybrid algorithm such a fine discretization would have been extremely time consuming and out of the question for an extension to many-body targets. In contrast, the calculations in Fig. 15 took less than 60 minutes with the cluster method. (All calculations discussed in this section were preceded by 50000 equilibration steps before starting with the evaluation of the expectation value of the modified observable; the quoted computation times always include this initial equilibration.) That the refined discretization was essential for the accurate result in Fig. 15 can be seen in Fig. 16, where different choices for  $\varepsilon$  are compared. Comparison of Figs. 11 and 16 shows that the influence of the discretization of the projection time is equal for both methods; a calculation with the same projection time  $\beta$  and discretization  $\varepsilon$  yields results of comparable quality for both the hybrid method used in Fig. 11 and the cluster method.

To right away also test the limits of the cluster method, we used attractive potentials which, as we have seen before, put higher demands on the stochastic algorithms. The same qualitative results as for the repulsive potential in Figs. 15 and 16 were also found for weakly attractive potentials as long as the strength does not exceed  $U_0 = -0.3 \text{ fm}^{-1}$ . However, for potentials with a stronger attractive part the necessary projection time increases rapidly. This can be seen in Fig. 17, which shows the dependence of the accuracy on  $\beta$ . With the efficient cluster method one can afford to go to projection times as high as  $900 \text{ fm}$  to achieve good agreement with the exact result; a coarser discretization than for the repulsive potential could be used. The computation time with the cluster method was about 120 minutes for the  $\beta = 900 \text{ fm}$  case. Calculations with such a long projection time are impossible with the methods of Section 4 within the limits we set for computation time. The results shown in Fig. 13 for the same attractive potential and with a projection time  $\beta = 200 \text{ fm}$  required 180 minutes and show no signs of convergence to the exact result.

Closer inspection of Fig. 17 shows that even for the best result some fluctuation persist. This feature is even more evident in Fig. 18, where the attractive part of  $-0.75 \text{ fm}^{-1}$  required a very long projection time, resulting in a computation time of 180 minutes. As discussed above, for high projection times the middle of the path moves out of the potential region. Reflections occur increasingly on planes at large distances from the origin, causing the endpoints of the path to move out of the potential region. This leads a low acceptance, slowing down the speed at which

uncorrelated paths are generated and fluctuations remain in the result.

To give a general impression of the performance of the cluster algorithm, we show in Fig. 19 a set of results for attractive and repulsive potentials with varying imaginary parts. The corresponding parameters are listed in Table 1. After the equilibration, 300 000 new paths were generated in all cases to evaluate the observable. The error-bars in Fig. 19 are obtained from the results of the last 50 000 steps. The solid curve shows the exact result obtained by numerically solving the Schrödinger equation with a Runge-Kutta method. The computation time for the repulsive potentials was 45 minutes, while the attractive potentials require a longer projection time  $\beta$  and took ca. 60 minutes each. The overall performance is quite good. The only significant discrepancies occur for the attractive potential, which is close to having a bound state and requires longer projection times. The examples with a weak imaginary potential could actually have been carried out in about half the computing time, since the scattering wave function of the full problem is quite similar to that with the real part of the potential only (which is used to generate the paths) and the convergence is fast.

The strength of the real simulation potentials we have used here varied from  $U_0 = -0.75 \text{ fm}^{-1}$  to  $2.3 \text{ fm}^{-1}$ . To put this range into perspective, we show in Fig. 20 the corresponding exact scattering length as a function of the strength parameter  $U_0$ . The scattering length changes substantially and grows rapidly as the real part of the scattering potential approaches  $U_0 = -1.2 \text{ fm}^{-1}$  where a bound state appears and  $f_0$  diverges.

In summary, the performance of the cluster algorithm is superior to that of any of the other algorithms discussed before. While the hybrid algorithms can be improved by finer discretization, the necessary computation time grows roughly quadratically with the number of points [19] and becomes unreasonably large. In contrast, the increase in computation time is only linear for the cluster algorithm, which allows one to obtain results quite efficiently even for the difficult cases of attractive potentials which the hybrid algorithm cannot manage anymore. In cases where both hybrid and cluster methods could be applied, the achieved accuracy was comparable. To further improve the successful cluster method and understand its performance in more detail, one would need to better understand the decorrelation of paths in this method. Further studies should also be directed towards optimizing the orientation of the reflection plane to yield a uniform acceptance of reflections of both the middle segments and end points of a path.

## 6 Summary

The path integral approach offers an opportunity to solve exactly the quantum mechanical scattering of a hadron from a composite, many-body target. The standard methods, which have been used for decades, involve several approximations concerning both the reaction mechanism between projectile and target as well as the internal target dynamics. Recent studies have shown that due these approximations one misses important physics, such as the trapping of the projectile by several scattering centers.

The possibility to apply the path integral formalism to low energy scattering has been pointed out several years ago. However, little has been done since then to actually use it in practice and to investigate which computational method is best suited for its application. The path integral approach is of course a common tool for the study of localized, bound systems. But when considering scattering one is faced with new aspects. An obvious difference is that in scattering the wavefunction is not normalizable and the spectrum of the Hamiltonian continuous. Another important feature that has to be dealt with by the numerical methods is the presence of two very different length scales. One is provided by the size of the target or the relatively short range of the interaction between projectile and the target constituents. The other scale, which is much larger, is introduced by the exterior region into which the paths can drift off with increasing physical time.

In order to assess the efficiency of the different methods for applying the path integral approach to scattering, we have tested them for a situation where the exact answer can still be obtained by standard methods: the scattering of a particle from a potential. As the idea is to find a method that can be extended to the much more extensive and time consuming calculations for a many-body target, the desired computational method has to be fast and numerically stable. Keeping in mind the applications to systems with absorptive channels, *e.g.* for calculating the antiproton - nucleus scattering length, we have used a complex potential. Due to the presence of a complex action, the usual probability interpretation of the distribution of paths is not possible. We therefore used the fact that one can choose a different real action to generate the paths and to accordingly rearrange the matrix-element of the observable one evaluates. In our approaches, the imaginary part of the potential is always contained in such a 'modified observable'. In the action used for generating paths we used here for simplicity the real part of the scattering potential; in other

applications it is of course possible to make another choice suggested by the physics of the system.

We compared three different methods to generate paths for the evaluation of the observable. The first is the random walk method, where paths are built up point for point. In contrast, the Langevin algorithm, with its extensions to the Langevin algorithm with a kernel and the hybrid algorithm, deals with the updating of entire paths, rather than stepping through each path in the physical time. These two methods, which had previously been applied to bound systems, showed systematic difficulties when applied to scattering. We could trace the difficulties of these methods, which have been used extensively for bound states, to the different physical situation in scattering and tested some modifications and hybrid methods. The insights gained from these studies finally this led us to propose the 'cluster algorithm' for scattering. It deals not with individual points of a path but instead with clusters of varying sizes.

For the random walk method we considered both path generation by the simple free diffusion method as well as the method of path replication. Both yielded rather inaccurate results, mainly due to the fact that only few of the generated paths actually contributed significantly to the integral. Some improvement was obtained by the 'time-dependent guided random walk', but large fluctuations in the observable persisted, in particular for attractive potentials, which require a large projection time and put higher demands on numerical stability.

While the random walk sequentially generates the elements of a path, the Langevin algorithm updates an entire path. As discretization errors become important, the generated paths were treated as proposals for a Metropolis procedure. Within the computational limits we considered, this procedure was seen to be quite time-consuming and yielded only unsatisfactory results. This was traced in part to the small number of uncorrelated paths produced in this fashion. To speed up the algorithm, a kernel was introduced into the stochastic differential equation, which allows one to decouple the long-ranged and short-ranged modes and substantially lowers the 'decorrelation time' between independent paths. This yielded very stable results, but only over a limited range of potential parameters. As the matrix algebra makes this method more time consuming, its application to more complicated scattering problems does not look promising. A similar performance was found when combining the Langevin method with deterministic methods to the 'hybrid algorithm'. This was shown to lead to a substantial gain in the the decorrelation time. Nevertheless,

the need for refined discretization also makes this method rather time consuming, especially when attractive potentials are considered.

Instead of dealing with individual points when updating a path, the 'cluster algorithm', which we finally proposed, treats groups of points as the new degrees of freedom. When generating a new path from a given one, these clusters are reflected collectively on a randomly oriented plane. Due to the procedure for forming and reflecting these clusters, this was seen to lead to an updating in terms of large scale clusters in the exterior region, while small scale clusters are used in the potential region, allowing an accurate probing of the details of the potential. This algorithm was found to be very stable and efficient, able to handle all but the very strongly attractive potentials. Another aspect that makes this method look very promising for application to nuclear targets is the reflection procedure. It allows a path to move quickly over larger distances, *e.g.* between regions that are separated by repulsive potential barriers. If point-by-point updating were used, this would be extremely time consuming.

For none of the three types of algorithms have we tried to optimize specific computational aspects, such as the matrix manipulations for the Langevin algorithm with a kernel. Individual improvements are clearly possible and will cut down the computation time in each case. However, we believe that the qualitative differences among the different approaches which we have found are quite general and that the cluster approach is the one that should be pursued first when considering scattering from nuclear targets.

## Acknowledgements

We would like to thank Prof. F. Lenz for initiating this work and for his continued interest. Discussions with Prof. U. Wiese were the stimulus for the development of the cluster method. J.H.K. thanks the Institute für Theoretische Physik III for its hospitality during several extended stays. The work of H.R.M. and S.L. was supported in part by the DFG Graduiertenkolleg 'Starke Wechselwirkung' Erlangen-Regensburg and by the Bundesministerium für Bildung und Forschung (BMBF). The work of J.H.K. is part of the research program of the Foundation for Fundamental Research of Matter (FOM) and the National Organisation for Scientific Research (NWO).

## References

- [1] F. Lenz and D. Stoll, in: Computational Nuclear Physics, Vol. 2, K. Langanke, J.A. Maruhn and S.E. Koonin, eds. (Springer, New York, 1993).
- [2] S. Lenz and D. Stoll, Z. Phys. A 351 (1995) 19.
- [3] S. Lenz, H. Mall, Comp. Phys. Comm. 90 (1995) 215.
- [4] R. H. Swendsen and J.-S. Wang, Phys. Rev. Lett. 58 (1987) 85.
- [5] U. Wolff, Phys. Rev. Lett. 62 (1989) 361.
- [6] M. Creutz, in: Quantum Fields on the Computer, Advanced Series on Directions in High Energy Physics, Vol. 11 (World Scientific, Singapore, 1992).
- [7] J. D. Doll et al., J. Chem. Phys. 89 (1988) 5753.
- [8] N. Makri, et al., J. Chem. Phys. 89 (1988) 2170.
- [9] C. Carraro and S. E. Koonin, Phys. Rev. B 41 (1990) 6741.
- [10] M. Lüscher, Commun. Math. Phys. 104 (1986) 153.
- [11] M. Lüscher and U. Wolff, Nucl. Phys. B339 (1990) 222.
- [12] I. Montvay and P. Weisz, Nucl. Phys. B290 (1987) 327.
- [13] H. Feshbach, Ann. Phys. (N.Y.) 164 (1985) 398.
- [14] M. A. Naimark, Linear Operators, Part II (Ungar, New York, 1968).
- [15] M. M. Sternheim and J. F. Walker, Phys. Rev. C 6 (1972) 114.
- [16] M. Kohno and W. Weise, Nucl. Phys. A45 (1986) 429.
- [17] C.J. Batty, in: Proceedings of the 12th Int. Conf. on Few Body Problems in Physics, Vancouver (1989).
- [18] R.P. Feynman and A. Hibbs, Quantum Mechanics and Path Integrals (McGraw-Hill, New York, 1965).
- [19] S. Lenz Dissertation, Univ. Erlangen (1994).

- [20] E. Onofri *et al.*, Physica Scripta 37 (1988) 323.
- [21] J. W. Negele, H. Orland, Quantum Many-Particle Systems (Addison-Wesley Publishing Company, 1988).
- [22] M. Namiki, Prog. Theor. Phys. Suppl. 111 (1993) 1.
- [23] K. Okano *et al.*, Prog. Theor. Phys. Suppl. 111 (1993) 313.
- [24] S. Tanaka *et al.*, Phys. Lett. B 288 (1992) 129.
- [25] A. D. Sokal, in: Quantum Fields on the Computer, Advanced Series on Directions in High Energy Physics, Vol.11 (World Scientific, Singapore 1992).
- [26] M. Creutz and A. Gocksch, Phys. Rev. Lett. 63 (1989) 9.
- [27] S. Duane, Nucl. Phys. B 257 (1985) 652.
- [28] S. Duane and J. B. Kogut, Nucl. Phys. B 275 (1986) 398.
- [29] S. Duane *et al.*, Phys. Lett. B 195 (1987) 216.
- [30] A. S. Kronfeld, Prog. Theor. Phys. Suppl. 111 (1993) 163.
- [31] H. G. Evertz *et al.*, Nucl. Phys. B (Proc. Suppl.) 20 (1991) 85.
- [32] U. Wiese, private communication



## Tables and Figures

$U_0$	$W_0$						
2.30	-0.50	-1.15	-2.30	-4.60	-7.00	.-	.-
-0.75	-0.02	-0.05	-0.15	-0.30	-0.50	-1.50	-2.25

Table 1: Potential parameters (in  $\text{fm}^{-1}$ ) used for Fig. 19.

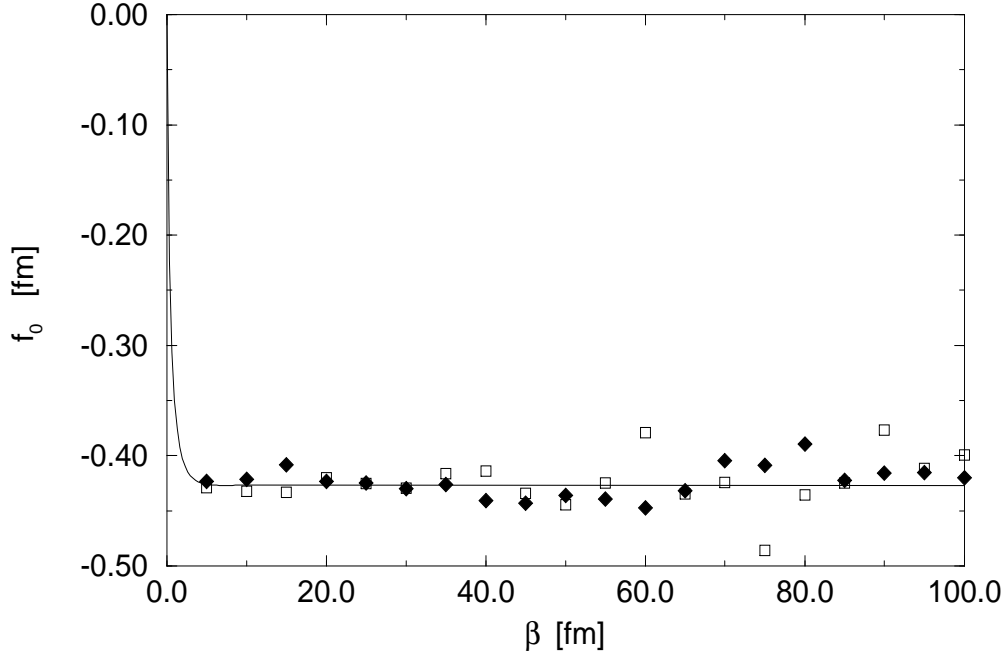


Figure 1: Scattering length  $f_0$  for repulsive potential,  $U_0 = 1.0 \text{ fm}^{-1}$ , as function of projection time,  $\beta$ ; discretization  $\varepsilon = 0.5 \text{ fm}$ . Open squares: with path replication; solid squares: free action random walk; solid curve: exact FFT result.

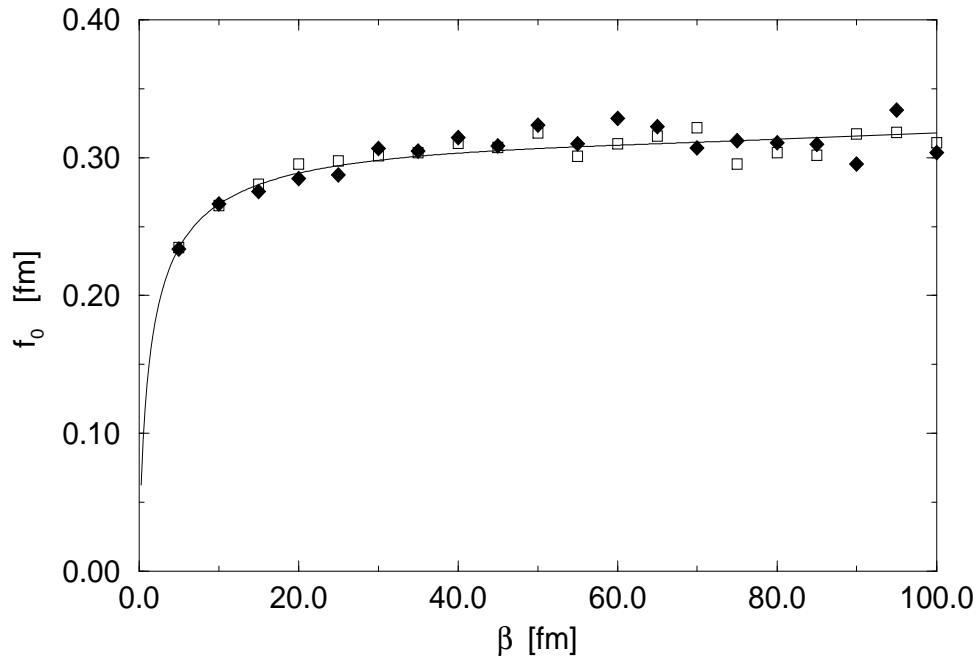


Figure 2: Scattering length  $f_0$  for attractive potential,  $U_0 = -0.3 \text{ fm}^{-1}$ , as function of projection time,  $\beta$ ; discretization  $\varepsilon = 0.5 \text{ fm}$ . Labelling as in Fig. 1

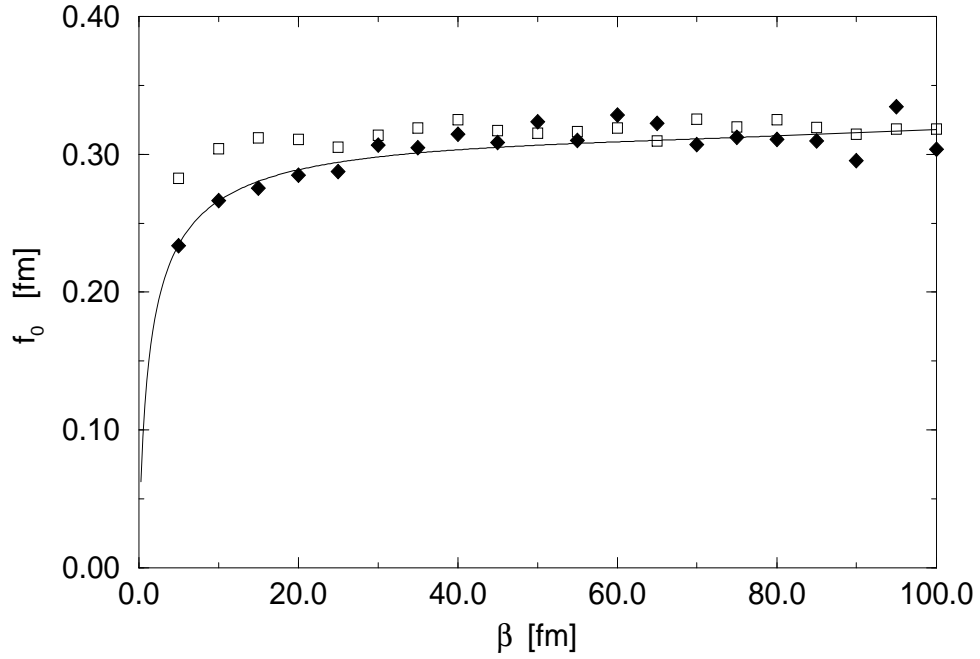


Figure 3: Scattering length  $f_0$  for attractive potential,  $U_0 = -0.3 \text{ fm}^{-1}$ , as function of projection time,  $\beta$ ; discretization  $\varepsilon = 0.5 \text{ fm}$ . Open squares: time dependent GRW; solid squares: free action random walk; solid curve: exact FFT result.

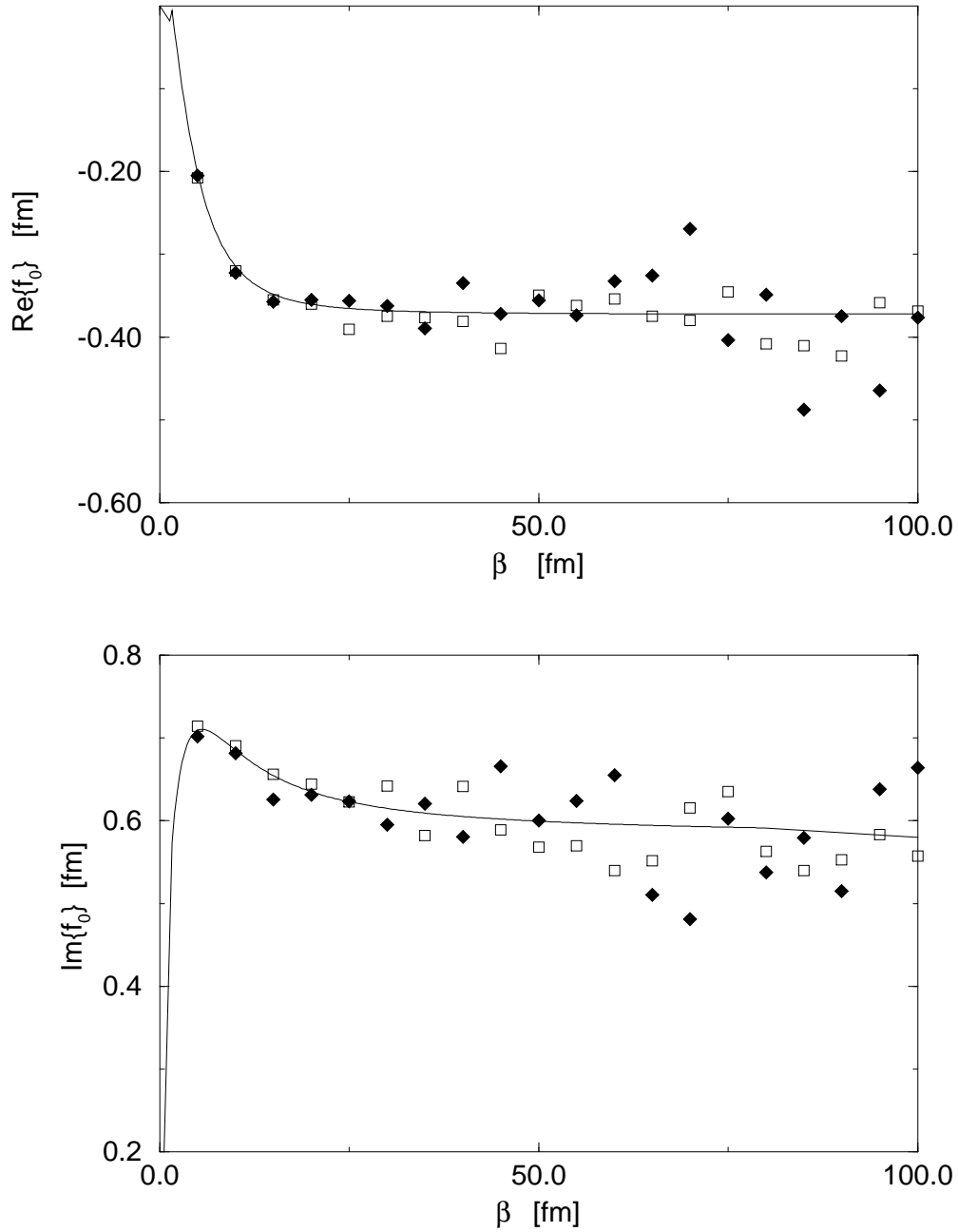


Figure 4: Scattering length  $f_0$  for complex potential  $V_c = (-0.3, -1.0) \text{ fm}^{-1}$  as function of projection time  $\beta$ ; discretization  $\varepsilon = 0.5 \text{ fm}$ . Open squares: time dependent GRW results; solid squares: free action random walk; solid curve: exact FFT result.

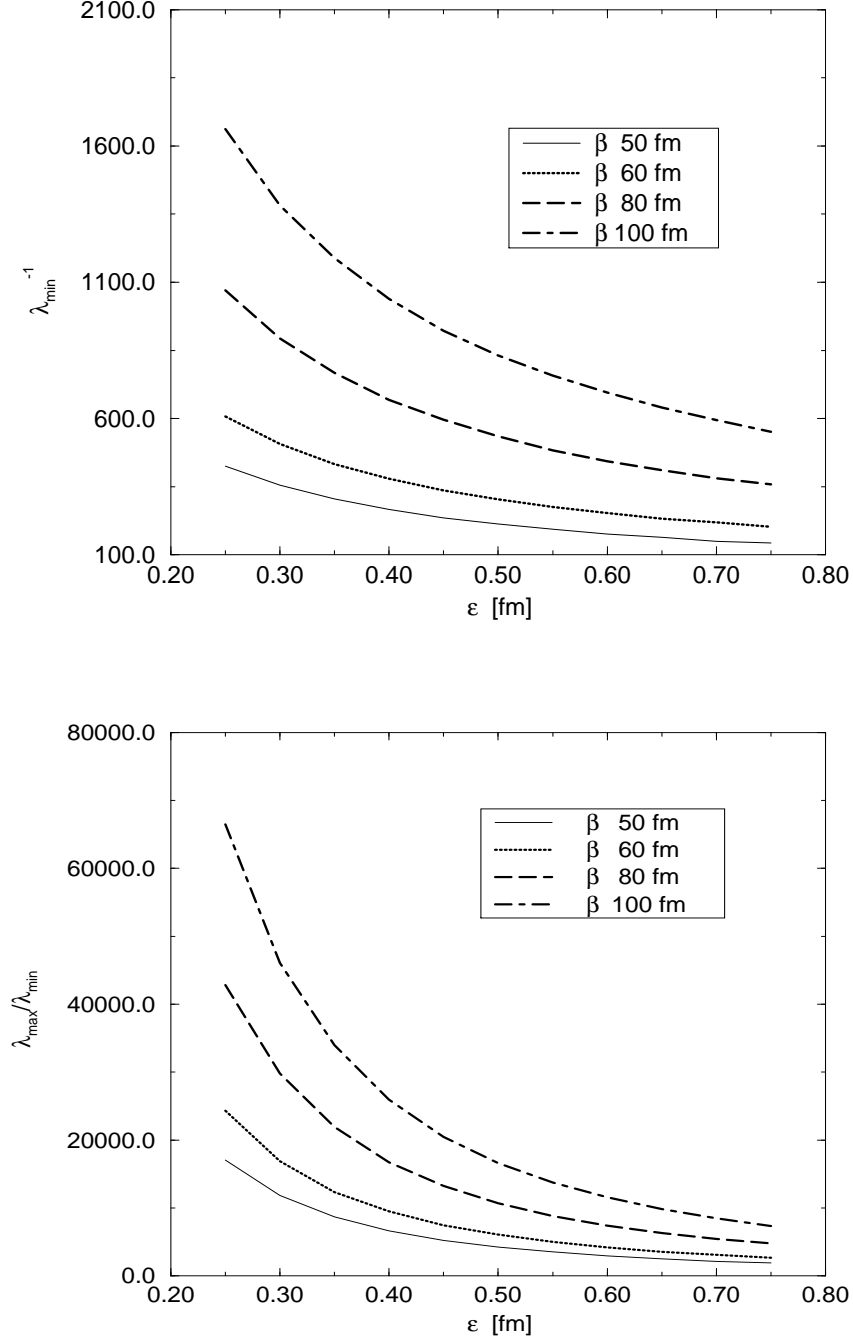


Figure 5: Upper graph: smallest eigenvalues of free action matrix  $\mathbf{A}_0$  as function of discretization  $\varepsilon$  for different projection times  $\beta$ . Lower graph: ratio of largest and smallest eigenvalue of  $\mathbf{A}_0$ .

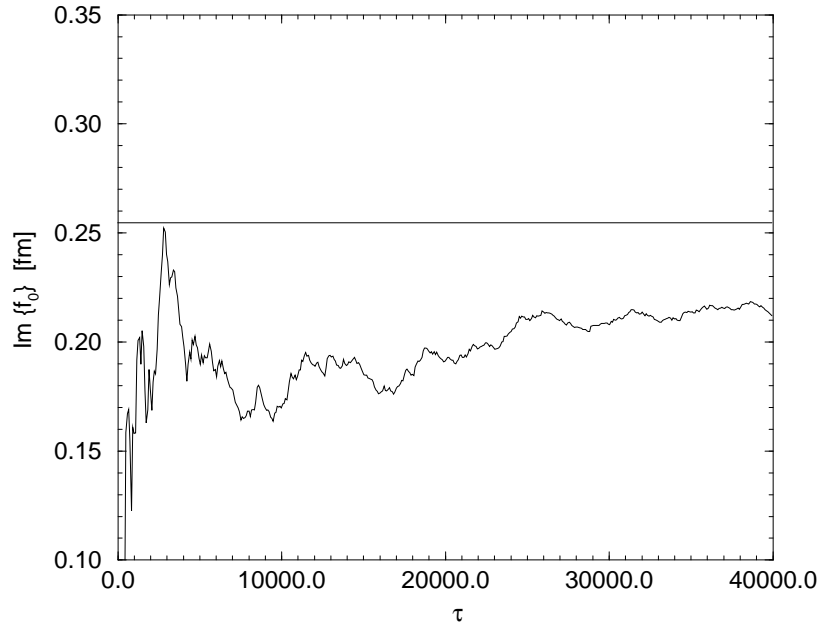
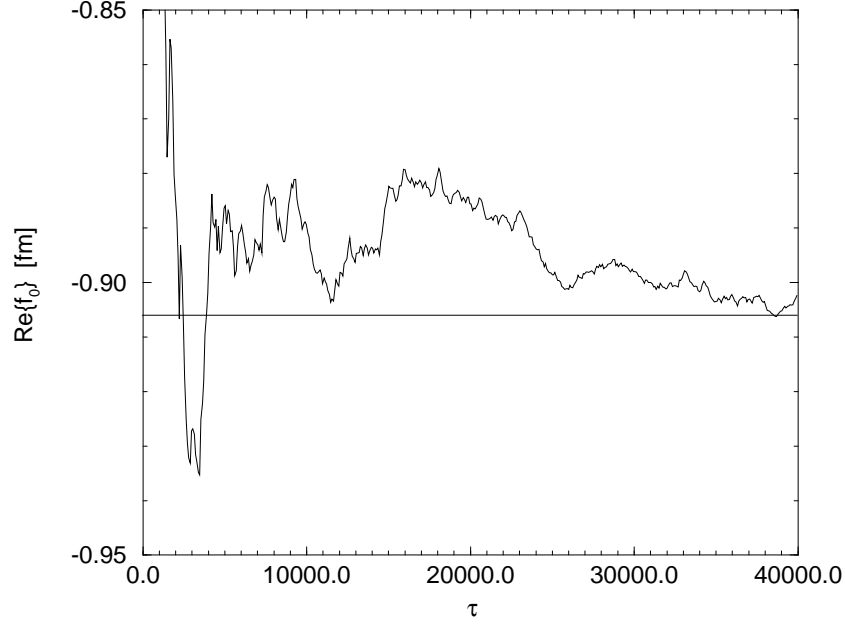


Figure 6: Scattering length for complex potential  $V_c = (2.3, -7.0) \text{ fm}^{-1}$ , obtained from the 'exact' Langevin algorithm;  $\beta = 50 \text{ fm}$ ,  $\varepsilon = 0.33 \text{ fm}$ . Horizontal line: exact result.

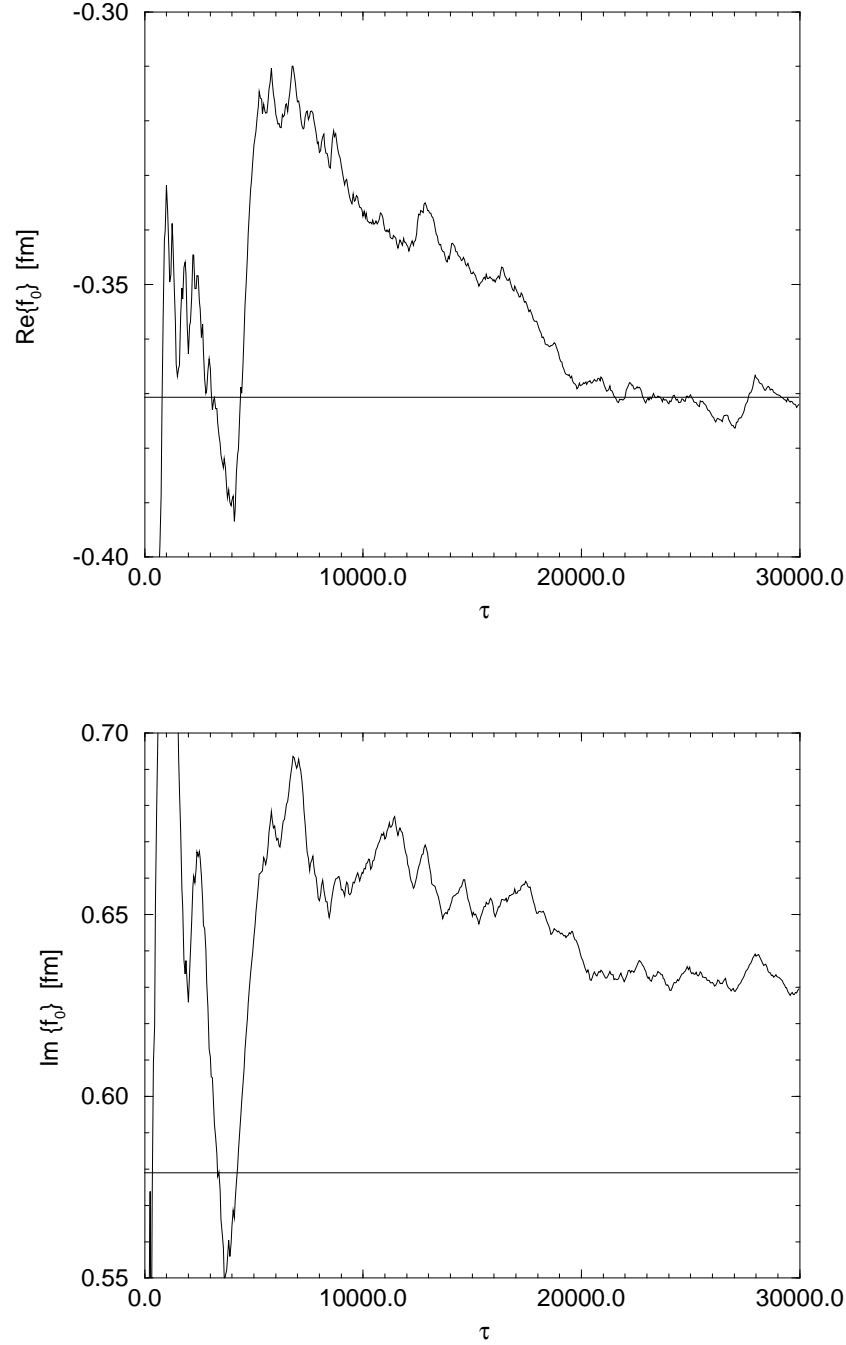


Figure 7: Scattering length for complex potential  $V_c = (-0.3, -1.0) \text{ fm}^{-1}$ , obtained from the 'exact' Langevin algorithm;  $\beta = 80 \text{ fm}$ ,  $\varepsilon = 0.5 \text{ fm}$ . Horizontal line: exact result.

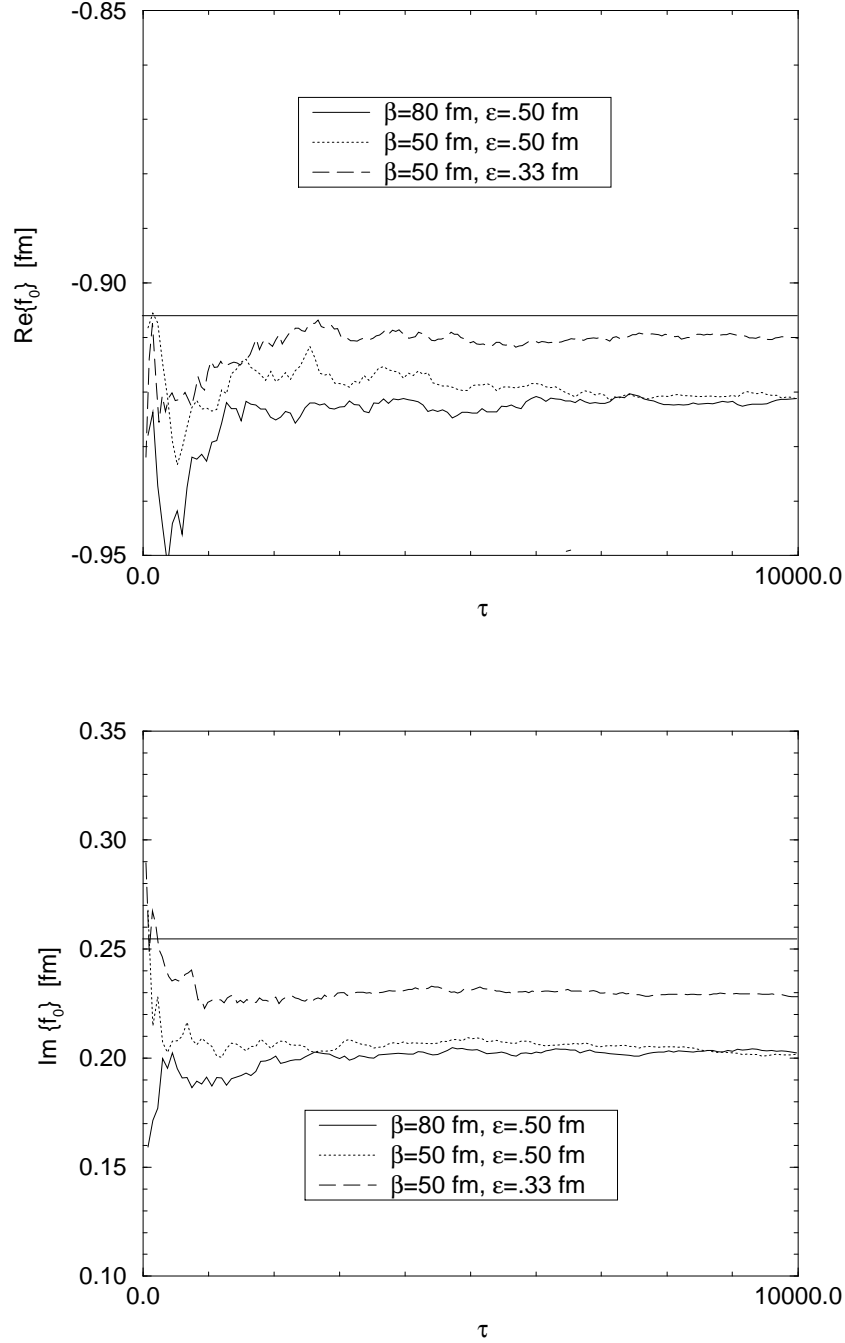


Figure 8: Scattering length for complex potential  $V_c = (2.3, -7.0) \text{ fm}^{-1}$ , obtained from the 'exact' Langevin algorithm with kernel. Horizontal line: exact result.



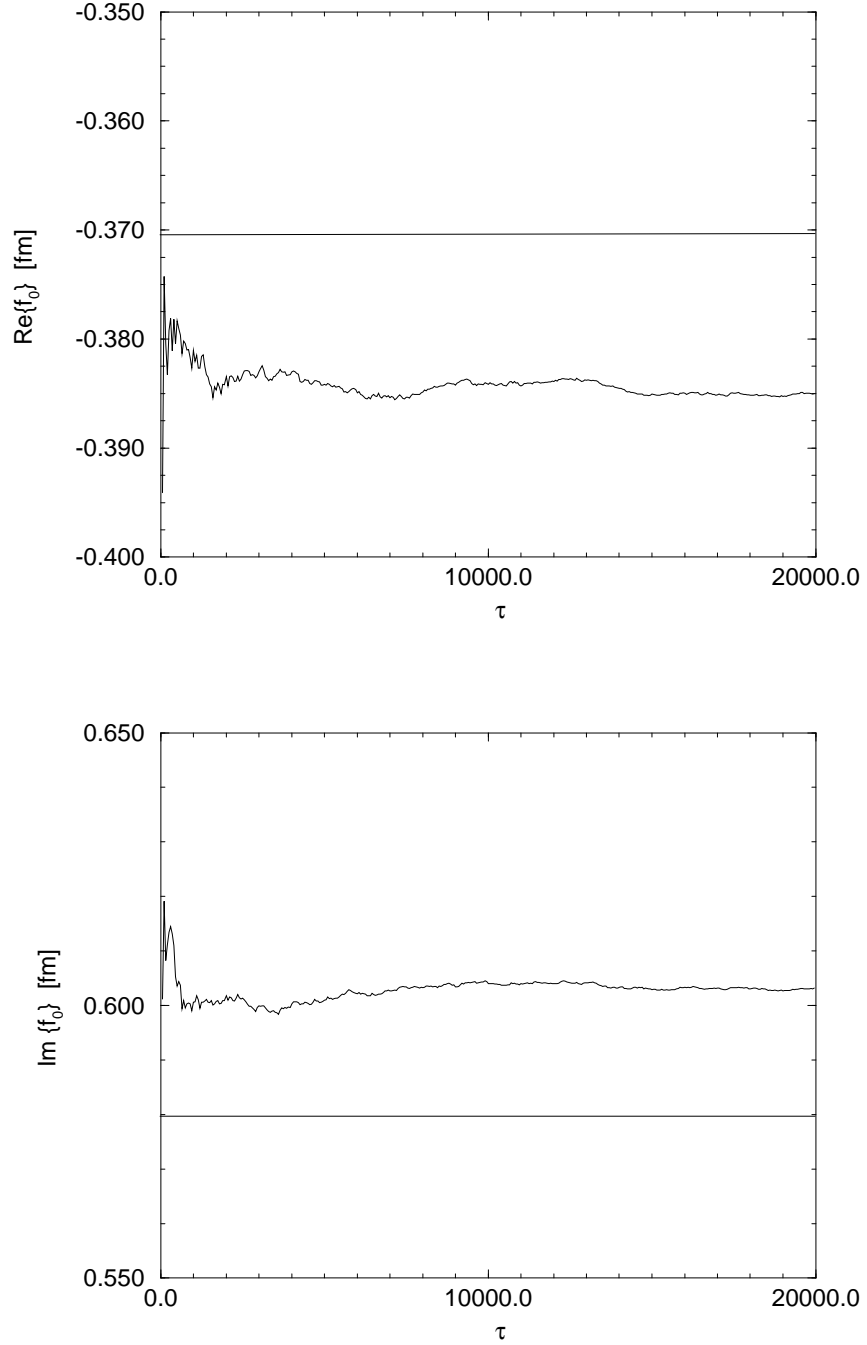


Figure 9: Scattering length for complex potential  $V_c = (-0.3, -1.0) \text{ fm}^{-1}$ , obtained from the 'exact' Langevin algorithm with kernel;  $\beta = 80 \text{ fm}$ ,  $\varepsilon = 0.5 \text{ fm}$ . Horizontal line: exact result.

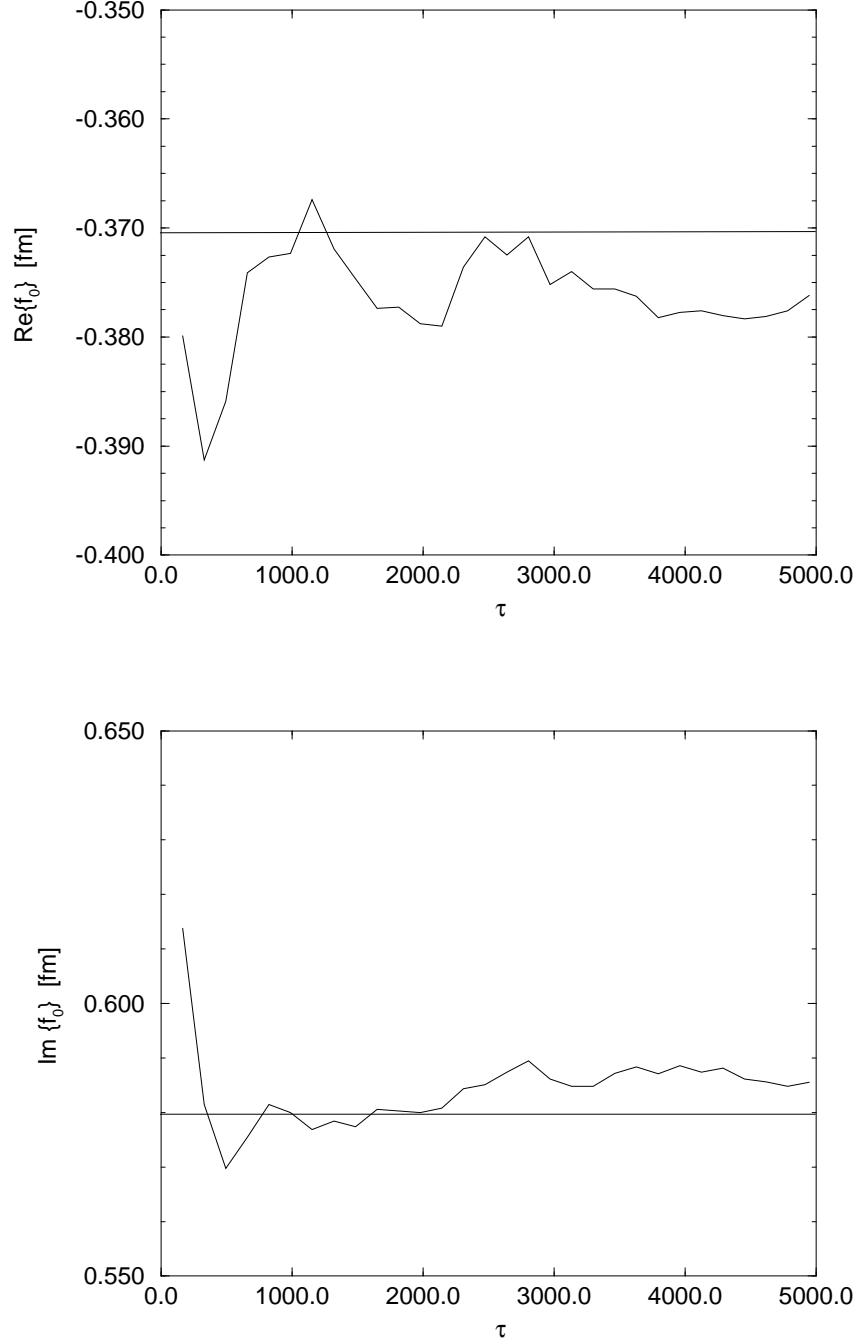


Figure 10: Scattering length for complex potential  $V_c = (-0.3, -1.0) \text{ fm}^{-1}$ , obtained from the 'exact' Langevin algorithm with kernel;  $\beta = 200 \text{ fm}$ ,  $\varepsilon = 0.5 \text{ fm}$ . Horizontal line: exact result.

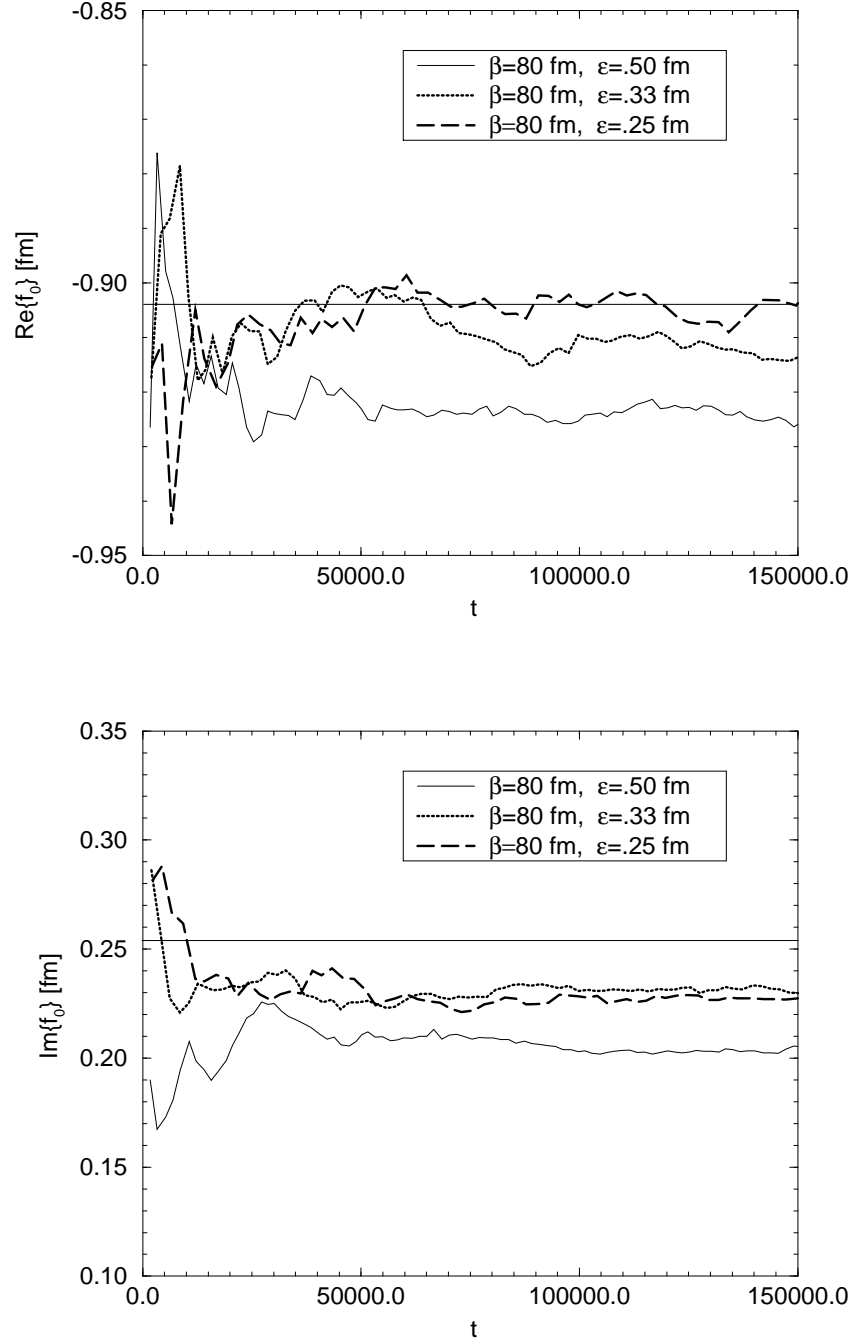


Figure 11: Scattering length for complex potential  $V_c = (2.3, -7.0) \text{ fm}^{-1}$ , obtained from the 'exact' hybrid algorithm. Horizontal line: exact result.

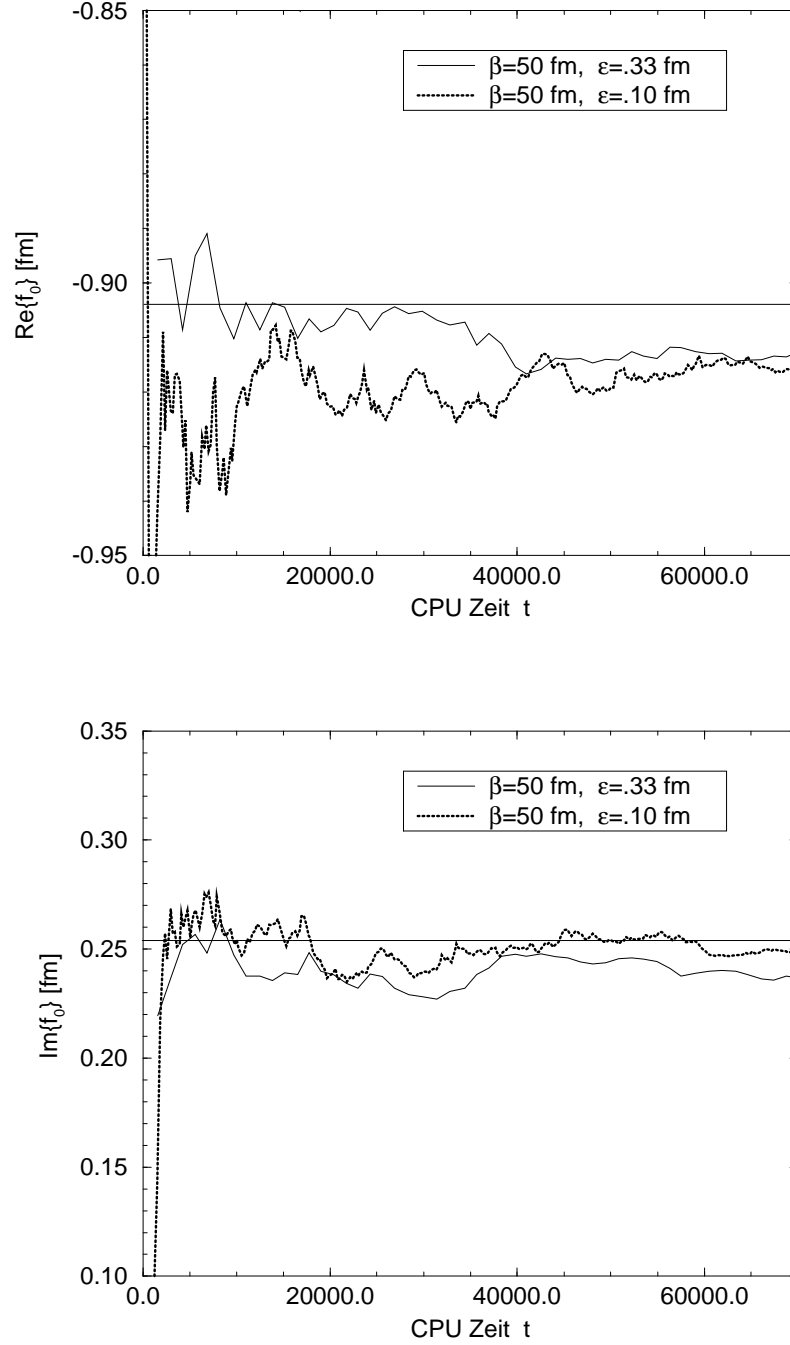


Figure 12: Scattering length for complex potential  $V_c = (2.3, -7.0) \text{ fm}^{-1}$ , obtained from the 'exact' hybrid algorithm. Horizontal line: exact result.

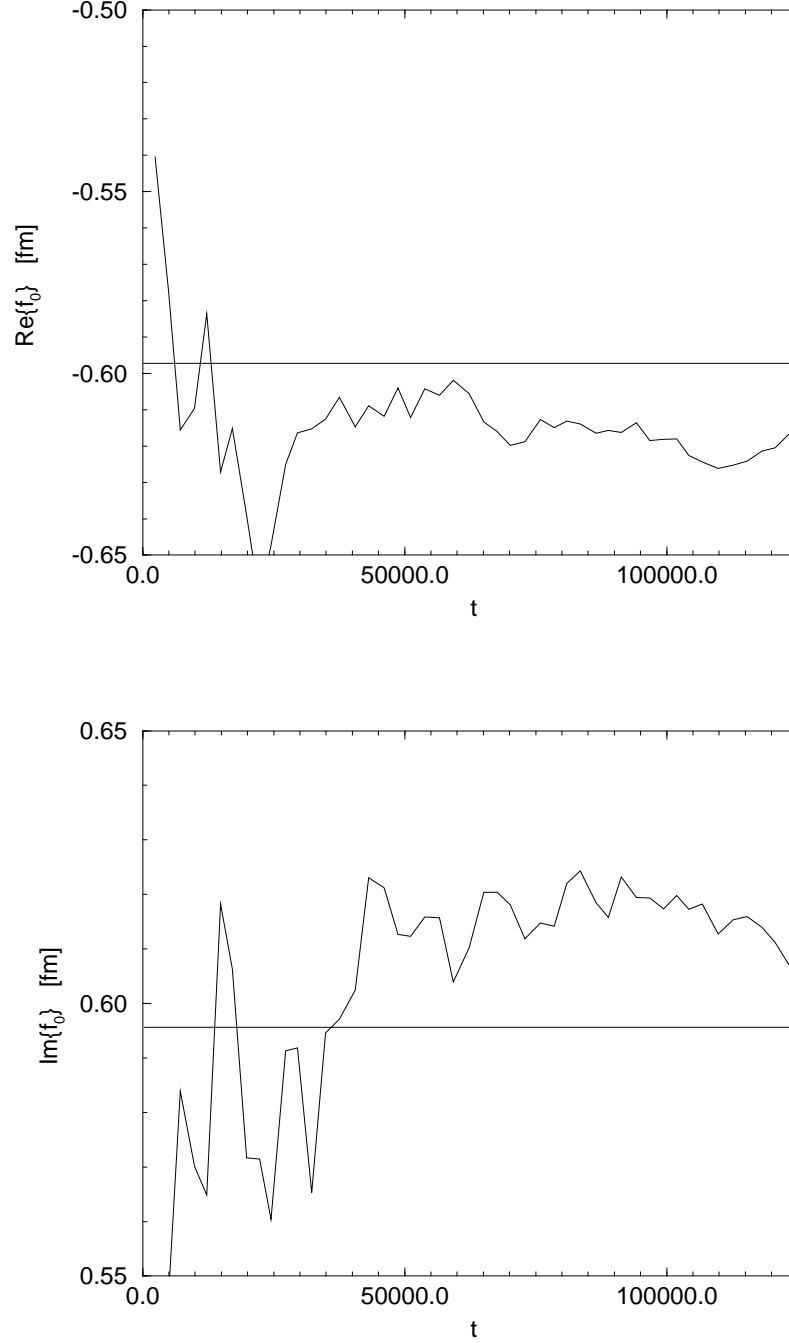


Figure 13: Scattering length for complex potential  $V_c = (-0.5, -1.5) \text{ fm}^{-1}$ , obtained from the 'exact' hybrid algorithm;  $\beta = 100 \text{ fm}$ ,  $\varepsilon = 0.33 \text{ fm}$ . Horizontal line: exact result.

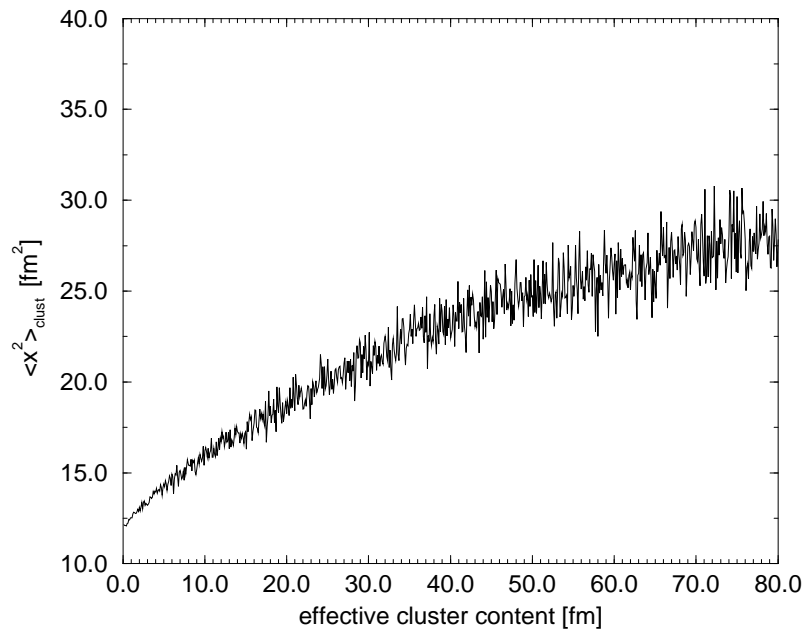


Figure 14: Mean square distance to the origin as function of effective cluster size for  $\beta = 80 \text{ fm}$ ,  $\varepsilon = 0.10 \text{ fm}$ ,  $V_c = (2.3, -7.0) \text{ fm}^{-1}$ .

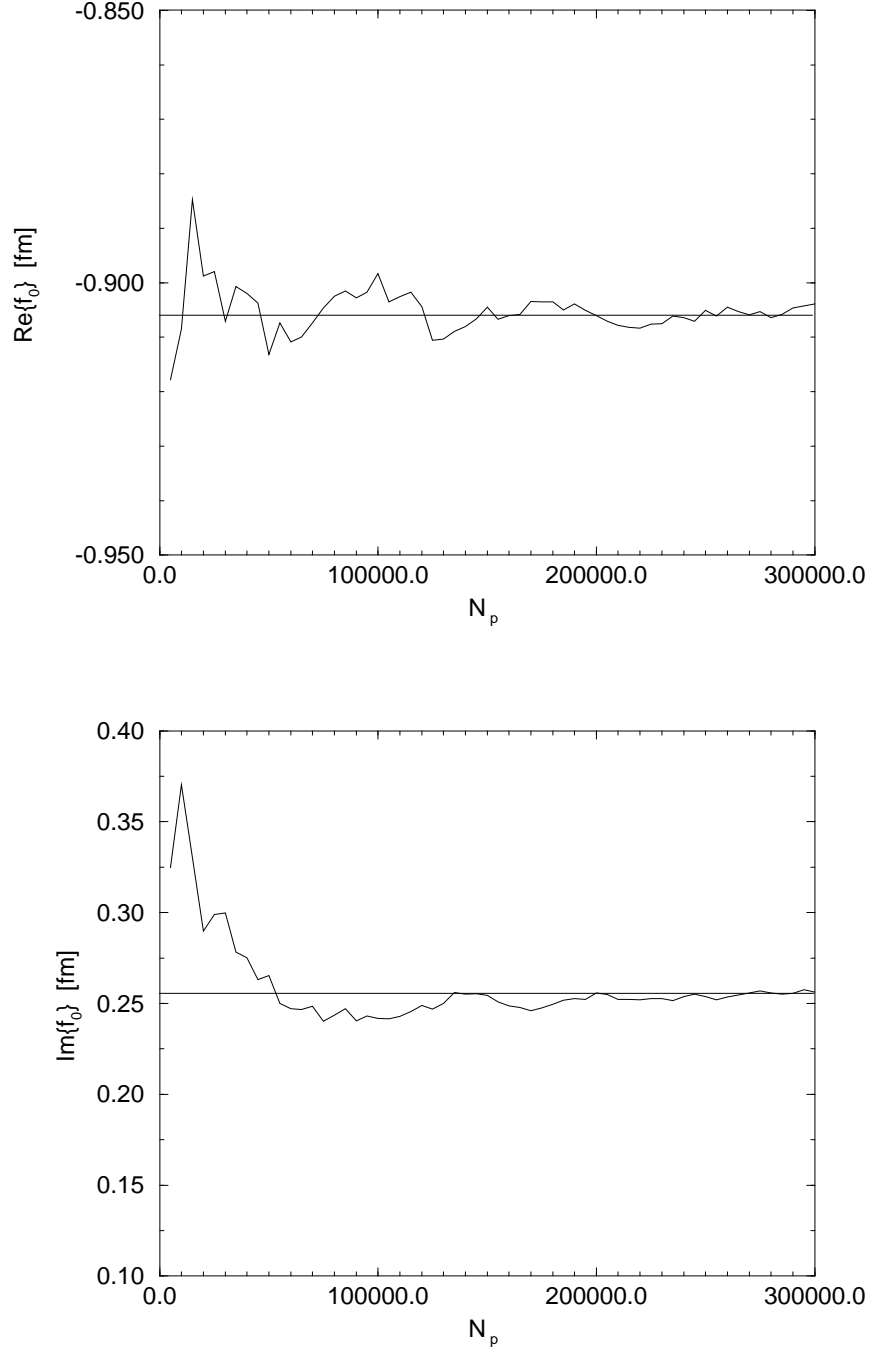


Figure 15: Scattering length  $f_0$  as function of the number of paths,  $N_p$ , generated by the cluster algorithm for  $V_c = (2.3, -7.0) \text{ fm}^{-1}$ ,  $\beta = 80 \text{ fm}$ ,  $\varepsilon = 0.10 \text{ fm}$ . Solid line: exact result.

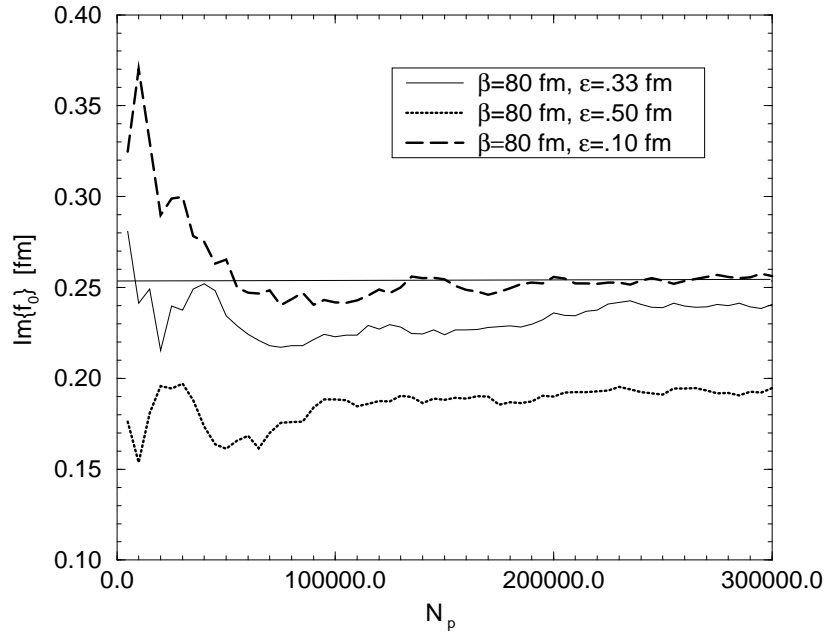
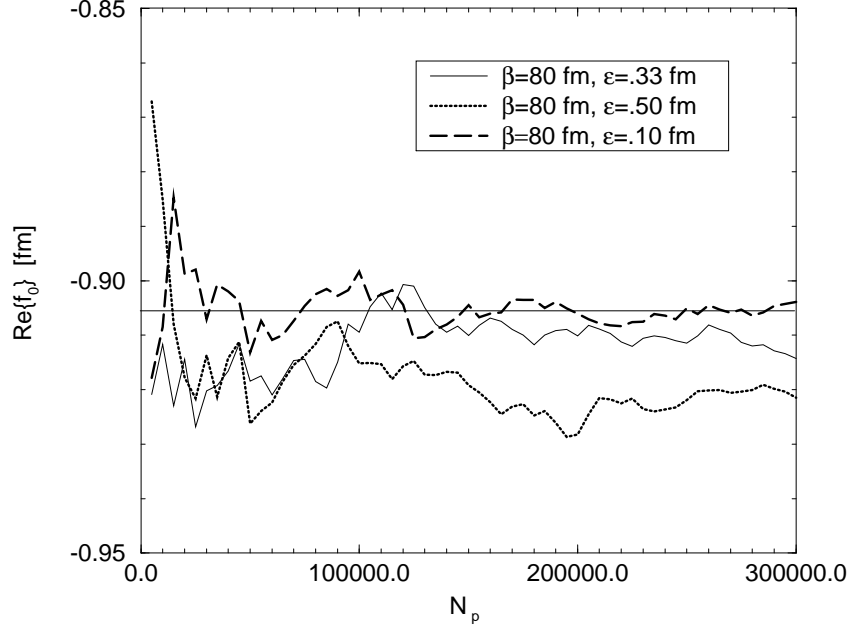


Figure 16: Scattering length  $f_0$  as function of the number of paths,  $N_p$ , generated by the cluster algorithm for  $V_c = (2.3, -7.0) \text{ fm}^{-1}$ ,  $\beta = 80 \text{ fm}$  and different discretizations,  $\epsilon$ . Solid line: exact result.



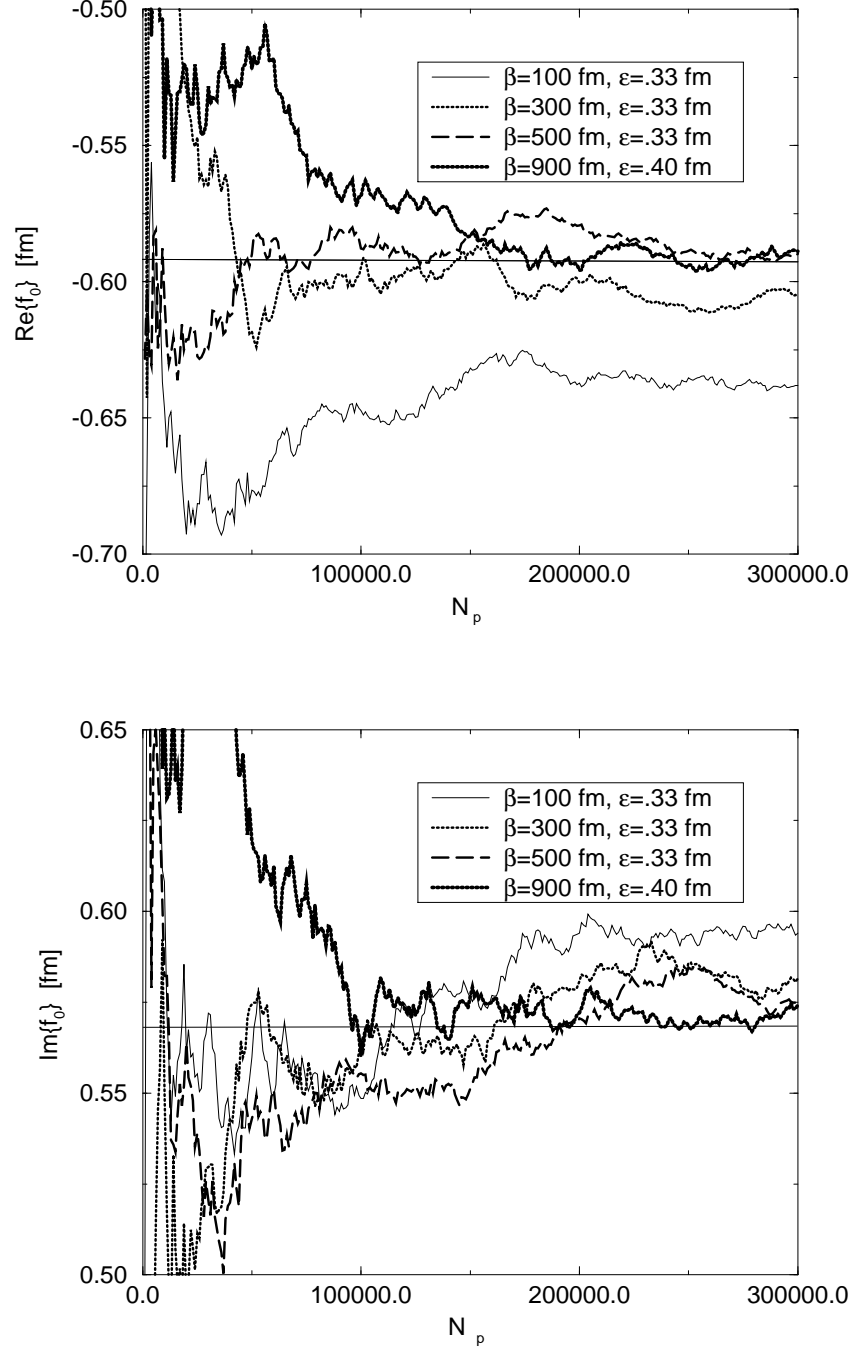


Figure 17: Scattering length  $f_0$  as function of the number of paths,  $N_p$ , generated by the cluster algorithm for  $V_c = (-0.5, -1.5) \text{ fm}^{-1}$ , different discretizations,  $\epsilon$ , and projection times,  $\beta$ . Solid line: exact result.

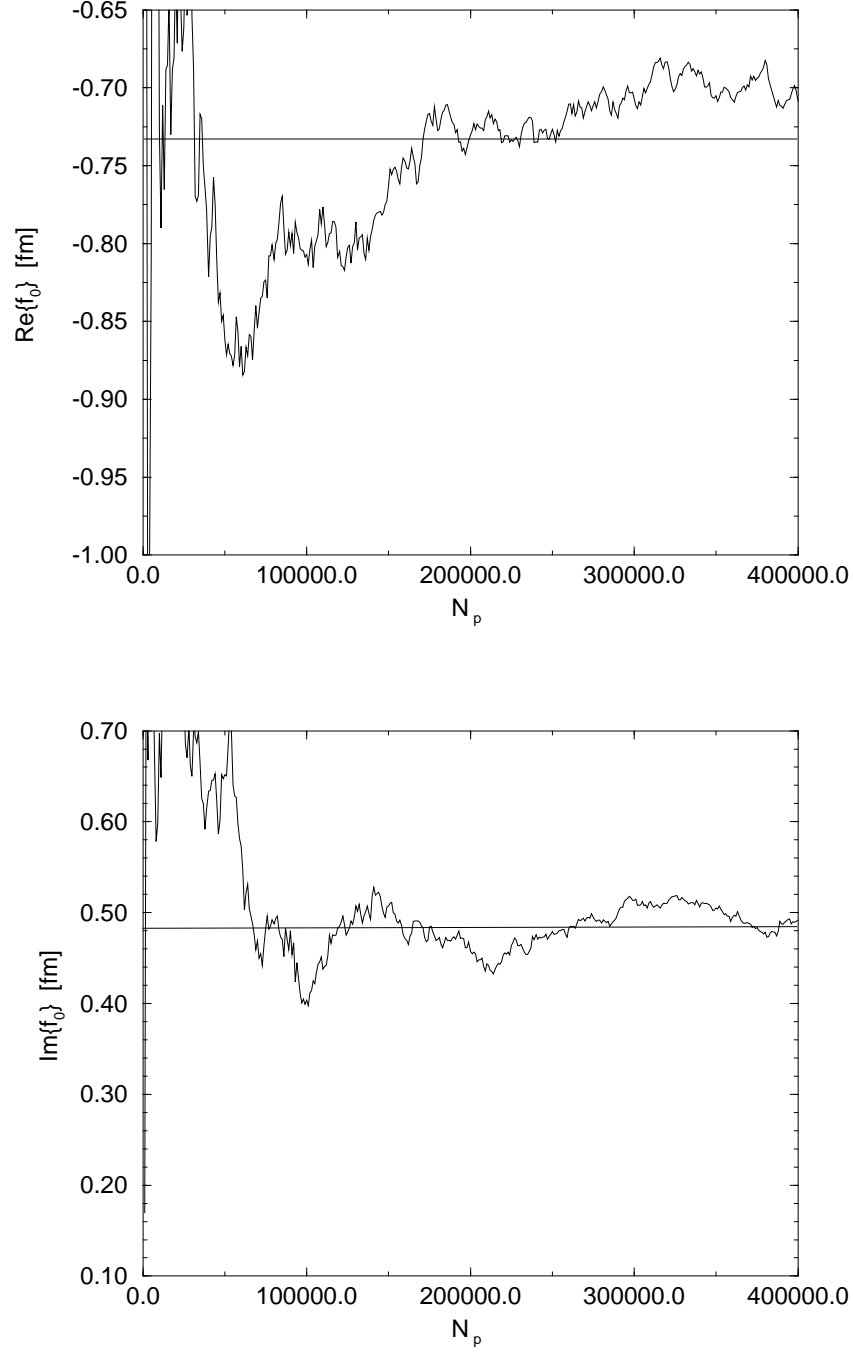


Figure 18: Scattering length  $f_0$  as function of the number of paths,  $N_p$ , generated by the cluster algorithm for  $V_c = (-0.75, -2.25) \text{ fm}^{-1}$ ,  $\beta = 900 \text{ fm}$ ,  $\varepsilon = 0.50 \text{ fm}$ . Solid line: exact result.

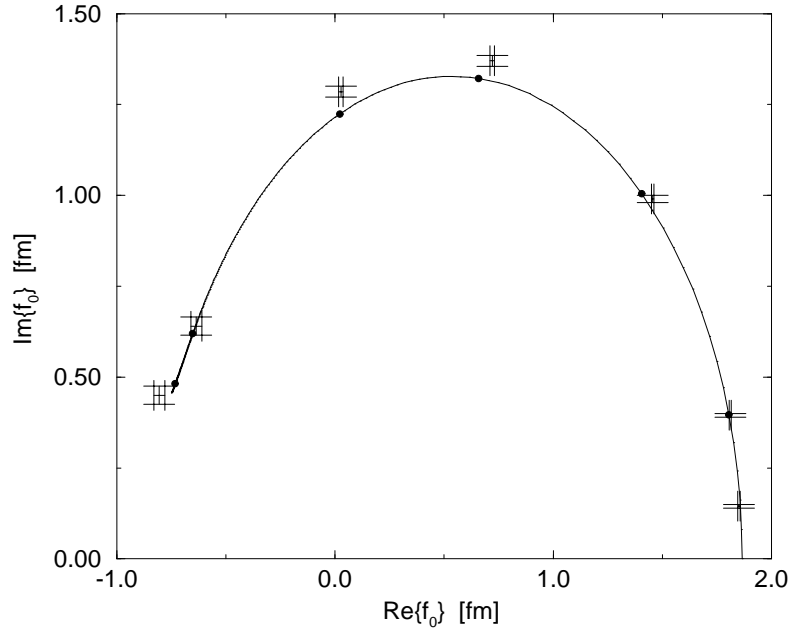
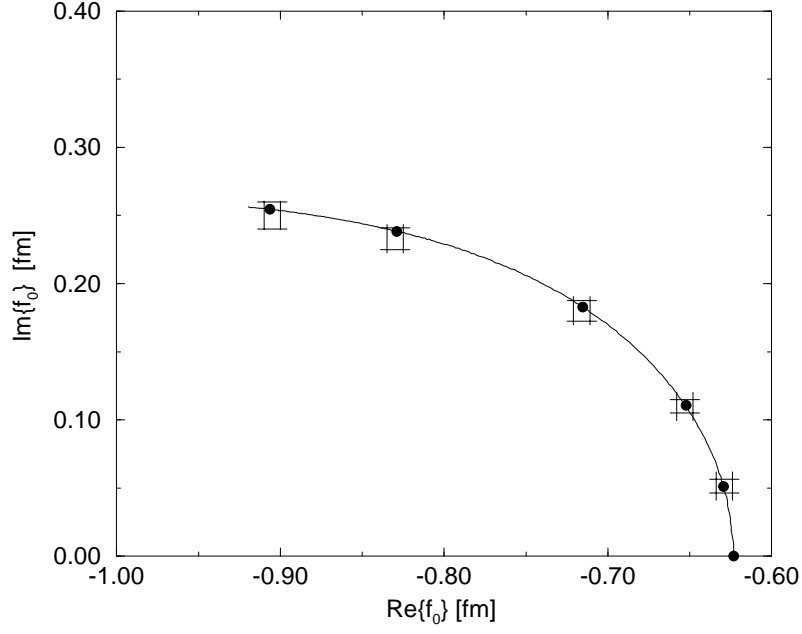


Figure 19: Scattering length for complex potentials with different absorptive parts given in Table 1; absorptive part grows in absolute value from right to left. Upper figure: attractive real part; lower figure: repulsive real part.

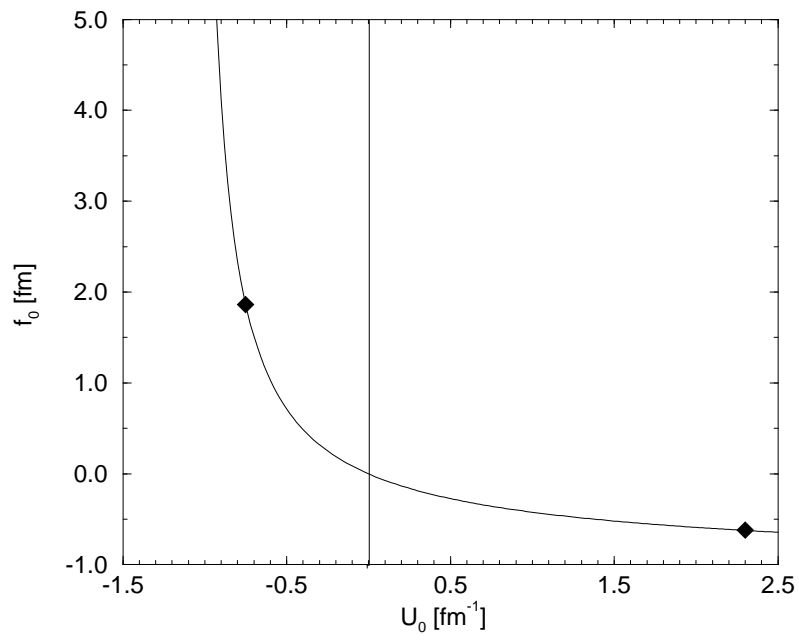


Figure 20: Scattering length  $f_0$  as function of strength parameter  $U_0$ . Range of real simulation potentials used for path generation indicated by diamonds.